Graph-based approaches for semi-supervised and cross-domain sentiment analysis

Natalia Ponomareva

A thesis submitted in partial fulfilment of the requirements of the University of Wolverhampton for the degree of Doctor of Philosophy

2014

This work or any part thereof has not previously been presented in any form to the University or to any other body whether for the purposes of assessment, publication or for any other purpose (unless otherwise indicated). Save for any express acknowledgements, references and/or bibliographies cited in the work, I confirm that the intellectual content of the work is the result of my own efforts and of no other person.

The right of Natalia Ponomareva to be identified as author of this work is asserted in accordance with ss.77 and 78 of the Copyright, Designs and Patents Act 1988. At this date copyright is owned by the author.

Signature: . . . . . .

Date: . . . . . .

Посвящается моей любимой мамулечке

Dedicated to my beloved mamulechka

Abstract

The rapid development of Internet technologies has resulted in a sharp increase in the number of Internet users who create content online. User-generated content often represents people's opinions, thoughts, speculations and sentiments and is a valuable source of information for companies, organisations and individual users. This has led to the emergence of the field of sentiment analysis, which deals with the automatic extraction and classification of sentiments expressed in texts. Sentiment analysis has been intensively researched over the last ten years, but there are still many issues to be addressed. One of the main problems is the lack of labelled data necessary to carry out precise supervised sentiment classification. In response, research has moved towards developing semi-supervised and cross-domain techniques. Semi-supervised approaches still need some labelled data and their effectiveness is largely determined by the amount of these data, whereas cross-domain approaches usually perform poorly if training data are very different from test data. The majority of research on sentiment classification deals with the binary classification problem, although for many practical applications this rather coarse sentiment scale is not sufficient. Therefore, it is crucial to design methods which are able to perform accurate multiclass sentiment classification.

The aims of this thesis are to address the problem of limited availability of data in sentiment analysis and to advance research in semi-supervised and cross-domain approaches for sentiment classification, considering both binary and multiclass sentiment scales. We adopt graph-based learning as our main method and explore the most popular and widely used graph-based algorithm, label propagation. We investigate various ways of designing sentiment graphs and propose a new similarity measure which is unsupervised, easy to compute, does not require deep linguistic analysis and, most importantly, provides a good estimate for sentiment similarity as proved by intrinsic and extrinsic evaluations.

The main contribution of this thesis is the development and evaluation of a graph-based sentiment analysis system that a) can cope with the challenges of limited data availability by using semi-supervised and cross-domain approaches b) is able to perform multiclass classification and c) achieves highly accurate results which are superior to those of most state-of-the-art semi-supervised and cross-domain systems. We systematically analyse and compare semi-supervised and cross-domain approaches in the graph-based framework and propose recommendations for selecting the most pertinent learning approach given the data available. Our recommendations are based on two domain characteristics, domain similarity and domain complexity, which were shown to have a significant impact on semi-supervised and cross-domain performance.

Acknowledgements

My PhD journey was quite long as it started away back in 2006 when I left Moscow to do a PhD at the Technical University of Valencia. Since then many things happened: I moved to the UK to work at the University of Wolverhampton, I changed the topic of my research, I left my studies in Valencia and started a new PhD in Wolverhampton, my dad passed away, I met my husband... Looking back at all these years I think that another valuable outcome of my thesis (apart from its significant research impact I hope :-)) is meeting during this journey many new friends who changed me in a better way and were always here to help and cheer me up. Thank you my dear friends and please excuse me if I do not mention you explicitly here, you are in my heart.

First of all, I would like to say a huge thank you to my supervisory team: Prof Mike Thelwall, Dr Mikhail Alexandrov and Dr Kevan Buckley. I feel lucky being supervised by Mike because he is not only a fantastic researcher but also one of the kindest and big-hearted people I have met in my life. Thank you Mike for giving me the opportunity to do a PhD under your supervision and for your valuable help and encouragements during all these years. I am very grateful to Mikhail Aronovich who motivated me to start a PhD in Spain and put a lot of effort to make it possible. He helped me

to sharpen my research skills and showed me an example of an enthusiastic researcher. I also want to thank him for being very supportive during hard personal times and caring about me and my family.

My studies in Valencia were marked by getting familiar with NLP and machine learning, meeting friends for life and lots of sun which I did not see much due to working all day. My passion for machine learning mostly started that time thanks to very informative lectures and seminars I attended there. For that I would like to say a big thank you to all professors at DSIC especially to Dr Paolo Rosso, Dr Ferran Pla, Dr Antonio Molina and Prof Francisco Casacuberta. I had a great time in Spain but it would not be the same without my lovely friends I met there: Jose Manuel, Chris and Jin.

Wolverhampton does not look like an exciting city to live but the place is all about people you meet there and I was very fortunate in that respect. In the first place, I would like to thank Alison and Fereshteh for being such great friends for me, for their love and care. I am grateful to my friends and colleagues at the University of Wolverhampton for making the PhD journey more pleasant and easier. I will not mention you all here, but you know that you are in my thoughts. A special thank you to my friends from the Russian Orthodox Church in Birmingham, and especially to father Mikhail, Alena and Irina, who encouraged me a lot during the last stages of my PhD. I am very grateful to Dr Laura Hasler for being very thorough and fast when proofreading my thesis and for her support and understanding during the last weeks before submission. I would like to thank my friends and colleagues from

Russia especially Masha, Dina, Tanya and my teacher Leila Alexandrovna who were always happy to hear from me and see me even after months of silence.

It would not be possible for me to complete this work without love and support of my family and especially two most important and beloved people in my life: my mum and my husband. Thank you mumulechka and Dinel for being with me! I will not say more because there are no words to express how much gratitude and love I feel towards you! I also want to thank my sister Tanya, Dashulya, Irisha, aunt Tamara and aunt Tanya. Unfortunately my dad did not live to see me a doctor but he believed in me so much when I started. I hope he would have been proud of me!

Contents

List of Tables **xiv**

List of Figures **xviii**

Abbreviations **xxiv**

1 Introduction **1**

1.1 Problem definition and learning approaches . . . 3

1.2 Graph-based learning . . . 6

1.3 Aims and goals . . . 8

1.4 Structure of the thesis . . . 10

2 Overview of research on sentiment classification **13**

2.1 Features . . . 16

2.2 Genres . . . 20

2.3 Approaches . . . 21

2.3.1 Lexical approaches . . . 22

2.3.2 Supervised classification . . . 25

2.3.3 Semi-supervised and unsupervised approaches . . . 30

2.3.4 Cross-domain learning . . . 36

2.4 Remarks on multiclass classification . . . 41

2.5 Summary . . . 42

3 Data, preprocessing and baselines **43**

3.1 Data . . . 44

3.2 Preprocessing . . . 46

3.3 Evaluation metrics . . . 50

3.4 Feature selection . . . 54

3.4.1 Related research . . . 55

3.4.2 Binary classification . . . 59

3.4.3 Multiclass classification . . . 66

3.4.4 Discussion . . . 71

3.5 Baselines . . . 73

3.5.1 Semi-supervised baselines . . . 73

3.5.2 Cross-domain baselines . . . 76

3.6 Summary . . . 80

4 Graph-based learning **81**

4.1 Notation and problem setting . . . 82

4.2 Sentiment graph construction . . . 85

4.2.1 Graph connectivity . . . 85

4.2.2 Sentiment similarity . . . 86

4.2.2.1 Document feature-based representation . . . 88

4.2.2.2 Document unit-based representation . . . 88

4.2.2.3 Adapting SO-CAL to review data . . . 90

4.2.2.4 Evaluation of similarity metrics . . . 94

4.3 Label Propagation . . . 97

4.4 Balancing class proportions . . . 99
4.5 LP modifications . . . 101
4.5.1 LP_γ: Weighting labelled and unlabelled neighbours . . . 102
4.5.2 $LP_{\alpha\beta}$: Incorporating external classifiers . . . 104
4.6 The design of the classification module . . . 107
4.7 Summary . . . 108
5 Study of data characteristics 109
5.1 Domain complexity . . . 111
5.2 Domain similarity . . . 114
5.3 Human annotation experiment . . . 119
5.3.1 Experiment description . . . 120
5.3.2 Evaluation . . . 122
5.3.2.1 Task complexity . . . 123
5.3.2.2 Human performance . . . 125
5.3.2.3 Inconsistency in review ratings . . . 127
5.4 Summary . . . 128
6 Semi-supervised experiments 131
6.1 Experimental setup . . . 132
6.2 LP and its modifications in the basic configuration . . . 134
6.3 The impact of normalisation and the hierarchical probability combination rule . . . 136
6.3.1 The binary case . . . 137
6.3.2 The multiclass case . . . 140

6.3.3 Discussion . . . 145
6.4 Sensitivity to parameter variations . . . 146
6.5 Extrinsic evaluation of similarity metrics . . . 153
6.5.1 The binary case . . . 154
6.5.2 The multiclass case . . . 156
6.6 The effect of the adapted SO-CAL dictionaries . . . 159
6.7 Comparison with other semi-supervised approaches . . . 161
6.8 Summary . . . 164
7 Cross-domain experiments **169**
7.1 Experimental setup . . . 171
7.2 The impact of normalisation and the hierarchical probability combination rule . . . 172
7.2.1 The binary case . . . 172
7.2.2 The multiclass case . . . 177
7.3 Sensitivity to parameter variations . . . 184
7.4 Analysis of the similarity measures . . . 187
7.5 Comparison with other cross-domain approaches . . . 189
7.6 Summary . . . 194
7.7 Semi-supervised vs. cross-domain graph-based learning . . . 197
8 Conclusions and future work **203**
8.1 Goals revisited . . . 203
8.2 Original contributions . . . 212
8.3 Directions for future research . . . 213

Bibliography **216**

A Identified sentiment markers **239**

B Amazon Review Coding Instructions **243**

B.1 Introduction 243

B.2 Filling in the annotation form 244

B.3 Rating judgements 244

C Previously published work **247**

List of Tables

2.1 Values of the parameters with which sentiment classification studies are categorised. 14

2.2 Research studies on sentiment classification. 15

3.1 Review corpora statistics. 47

3.2 Confusion table for the class C_k 50

3.3 20 most discriminative features from the music domain according to three feature ranking functions: IG, LR and WLLR. 59

3.4 Performance upper bounds for the binary and multiclass tasks. 73

3.5 20 discriminative positive features ranked by likelihood. 78

3.6 20 discriminative negative features ranked by likelihood. . . . 79

4.1 Distribution of the new sentiment markers over domains . . . 92

4.2 Evaluation of the similarity metrics based on different document representations ("+" adds the corresponding component to all those above it; the best combination is highlighted). 96

5.1 Pearson correlation between the domain complexity measures and in-domain accuracies given by SVM and VP. The data comprises the seven domains of different sizes: 25%, 50%, 75% and 100% of the whole amount of data in a domain, resulting in 28 data points altogether. 113

5.2 Correlation between various domain similarity measures and the cross-domain accuracies of SVM and VP calculated on 42 source-target domain pairs. 118

5.3 Distribution of the coder judgements over the 5* scale. 122

5.4 κ coefficients between coders C_1, C_2, C_3. 124

5.5 κ coefficients between coders C_1, C_2, C_3, gold standard C^* and reviewers U. 126

5.6 Percentage agreements between coders C_1, C_2, C_3, gold standard C^* and reviewers U. 126

5.7 Examples of label mismatches in product reviews. 128

6.1 Variations of $\bar{F}_1$ for LP_γ, $LP_{\alpha\beta}$ and $RANK$, different sizes of labelled data (100, 300 and 700 examples) and parameter ranges: α=200, $\beta \in \{0.2, 0.5, 1, 2, 5\}$, $k_u \in \{5, 10, 20, 50, 100\}$ and $\Delta_l \in \{0.05, 0.1, 0.2, 0.3, 0.4, 0.5\}$ (minimum values that surpass the B-line are highlighted, minimum values that surpass the U-bound are underlined). 148

6.2 Optimal parameter values for different configurations of LP_γ, $LP_{\alpha\beta}$ and $RANK$. 149

6.3 Document frequency of the new sentiment words and their average percentage out of all sentiment words in documents. . 161

6.4 Comparison of the results (accuracies) given by the feature-based and hybrid LP_γ+LB algorithms and four state-of-the-art semi-supervised approaches (the graph-based accuracies outperforming the best state-of-the-art results with a significance level of 0.05 are highlighted). 163

7.1 Optimal parameter values for LP_γ+$HIER$+LB and $RANK$+$HIER$. 185

List of Figures

3.1 Class distribution for the multiclass datasets. 45

3.2 Preprocessing steps 48

3.3 In-domain accuracies for the unigram and unigram+bigram vector model (binary case). 60

3.4 Feature weight impact on the in-domain accuracies (binary case)[1]. 61

3.5 Word form impact on the in-domain accuracies (binary case). 62

3.6 IG feature selection (binary case). 64

3.7 Comparison of feature selection techniques: IG, WLLR and LR (binary case). 66

3.8 Comparison of the feature selection with the full feature set results (binary case). 67

3.9 In-domain accuracies based on unigram and unigram+bigram vector representation (multiclass case). 68

3.10 Feature weight and word form impact on the in-domain accuracies (multiclass case). 69

3.11 IG feature selection (multiclass case). 70

3.12 Comparison of the feature selection with the full feature set results (multiclass case). 71

3.13 Semi-supervised baselines (binary case). 74

3.14 Semi-supervised baseline accuracies (multiclass case). 74

3.15 Semi-supervised baseline $macroF_1$ values (multiclass case). . . 75

3.16 Cross-domain accuracy baselines where each curve corresponds to the same target dataset (binary case). 76

3.17 Cross-domain accuracy baselines where each curve corresponds to the same target dataset (multiclass case). . . . 77

3.18 Cross-domain $macroF_1$ baselines where each curve corresponds to the same target dataset (multiclass case). . . . 77

4.1 Graph structure for LP 97

4.2 LP modifications: **A** Different weight for labelled and unlabelled neighbours (LP_γ); **B** Incorporating external predictions ($LP_{\alpha\beta}$). 103

4.3 The main stages of the classification module. 107

5.1 The relationship between complexity measures and in-domain accuracies given by SVM and VP. 114

5.2 The relationship between $-D_{JS}$ values and the cross-domain accuracies of SVM and VP, calculated on 42 source-target domain pairs. 118

6.1 Best accuracy averaged by domain for LP, LP_γ and $LP_{\alpha\beta}$ (binary case). 134

6.2 Best ($\bar{F}_1$) averaged over domains for LP, LP_γ and $LP_{\alpha\beta}$ (multiclass case). 136
6.3 Best accuracy averaged over domains for different algorithms and different normalisation techniques (binary case). 138
6.4 Best accuracy averaged over domains for the most successful algorithms and normalisation techniques (binary case). 139
6.5 Accuracy and MSE obtained with $LP_\gamma + LB$ for each domain (binary case). 139
6.6 Accuracy and $macroF_1$ averaged over domains for LP_γ, $LP_{\alpha\beta}$ and $RANK$ in different configurations (multiclass case). . . . 141
6.7 Best $\bar{F}_1$ averaged over domains for the most successful algorithms, normalisation techniques and probability combination rules (multiclass case). 142
6.8 Accuracy, $macroF_1$ and MSE obtained with LP_γ+LB and $RANK$+$HIER$ for each domain (multiclass case). 143
6.9 Sensitivity of LP_γ+LB, $LP_{\alpha\beta}$+$HIER$+LB and $RANK$+$HIER$ to variations of their parameters. The results are given for different sizes of labelled data: 100, 300 and 700 examples. . . 150
6.10 Sensitivity of $LP_{\alpha\beta}$+$HIER$+LB to variations of the parameter α. The results are given for 300 and 700 labelled examples. 152
6.11 The effect of different document representation components on the accuracy of LP_γ+LB (binary case). 157

6.12 The effect of different document representation components on the $\bar{F}_1$ of LP_γ+LB (multiclass case). 158

6.13 Percentage point differences between results with initial and adapted SO-CAL dictionaries. 160

7.1 Accuracy averaged over source domains with target domains fixed, for different algorithms and their configurations (binary case). 173

7.2 Accuracy averaged over source domains with target domains fixed, for the most successful algorithms and normalisation techniques (binary case). 174

7.3 Accuracy and MSE obtained with LP_γ+LB and $RANK$ for each source-target domain pair. X-axes contain source domains, graphs correspond to target domains (binary case). . 176

7.4 Accuracy obtained with LP_γ+LB and $RANK$ for each source-target domain pair. X-axes contain target domains, graphs correspond to source domains (binary case). 177

7.5 $\bar{F}_1$ averaged over source domains with target domains fixed, for different algorithms and their configurations (multiclass case). 178

7.6 Accuracy and $macroF_1$ averaged over source domains with target domains fixed for different configurations of LP_γ and $RANK$ (multiclass case). 179

7.7 Accuracy, $macroF_1$ and MSE obtained with LP_γ+$HIER$+LB and $RANK$+$HIER$ for each source-target domain pair. X-axes contain source domains, graphs correspond to target domains (multiclass case). 182

7.8 Accuracy, $macroF_1$ and MSE obtained with LP_γ+$HIER$+LB and $RANK$+$HIER$ for each source-target domain pair. X-axes contain target domains, graphs correspond to source domains (multiclass case). 183

7.9 Sensitivity of LP_γ+$HIER$+LB, and $RANK$+$HIER$ to variations of their parameters. The results are presented for groups of domains, compiled by their complexity. 186

7.10 The effect of different document representation components on the cross-domain results (multiclass case). 191

7.11 Comparison of the two best graph-based algorithms, LP_γ+LB and $RANK$, with state-of-the-art methods (accuracies). . . . 192

7.12 Comparison of the semi-supervised and cross-domain approaches (binary case). 199

7.13 Comparison of the semi-supervised and cross-domain approaches (multiclass case). 200

List of Abbreviations

ADN	Active Deep Networks	32
B-line	Baseline	134
BO	BOok reviews	44
BOW	Bag Of Words	16
CMN	Class Mass Normalisation	100
DV	movie reviews on DVDs	44
EL	reviews on ELectronic devices	44
GI	General Inquirer	24
HE	reviews on HEalth products	44
HIER	HIERarchical probability combination rule	99
IADN	Information Active Deep Networks	32
IG	Information Gain	17
JST	Joint Sentiment-Topic	35
KI	reviews on KItchen appliances	44
LB	Label Bidding	101
LDA	Latent Dirichlet Allocation	35
LP	Label Propagation	7
LR	Likelihood Ratio	57
maxP	maximum Probability combination rule	99

ME	Maximum Entropy	25
MSE	Mean Square Error	52
MU	reviews on MUsic	44
NB	Naive Bayes	25
NLP	Natural Language Processing	3
OR	Odds Ratio	56
PMI	Pointwise Mutual Information	22
PoS	Parts of Speech	16
PSP	Positive Sentence Percentage	33
PWP	Positive Word Percentage	89
RANK	graph RANKing	104
SCL	Structural Correspondence Learning	37
SFA	Spectral Feature Alignment	38
SVM	Support Vector Machine	17
SVR	Support Vector machine Regression	41
TitlePWP	Positive Word Percentage in Titles	89
TO	reviews on TOyes	44
TTR	Type/Token Ratio	46
U-bound	Upper bound	134
VP	Voted Perceptron	113
WLLR	Weighted Log-Likelihood Ratio	57

Chapter 1

Introduction

Sentiment analysis (Pang and Lee, 2008; Liu, 2012) has received lots of attention from the research community and industry for the last decade. This period was distinguished by the extremely fast development of Internet technologies, which led to their easy availability and mass exploitation. These factors enabled an immense growth of Internet users who create a vast amount of data each day. User generated content is a very valuable source of information, as it contains people's opinions and judgments on different topics and its automatic mining can be beneficial for companies, organisations and individual users.

Sentiment classification is one of the tasks within the sentiment analysis research field which is concerned with automatic identification of the sentiment strength or valence of texts (Pang et al., 2002; Turney, 2002; Gamon, 2004; Prabowo and Thelwall, 2009; Thelwall et al., 2010; Taboada et al., 2011; Bai, 2011). In spite of the research that has been carried out in this area recently, there still exist many issues that need to be addressed. One of the principal problems in sentiment analysis is the lack of labeled data to be able to conduct highly precise supervised classification. This

was the reason why the research moved towards developing semi-supervised and cross-domain techniques. However, semi-supervised approaches still need some labelled data and their effectiveness is largely determined by the amount of these data, while cross-domain approaches can produce poor results if training data are very different from test data. Usually researchers prefer one of these approaches and therefore there is a lack of studies on which approach is more beneficial for given data.

The majority of research on sentiment classification deals with the binary classification problem, when only positive and negative classes are being identified. Yet, for many practical applications (for example, for analysing customer satisfaction or the decision-making process when purchasing products or services) this rather coarse sentiment scale is not sufficient. A recent study discovered that consumers are "willing to pay from 20% to 99% more for a 5-star-rated item than a 4-star-rated item" (Pang and Lee, 2008, page 1). Therefore, it is crucial to design methods which are able to tackle the multiclass sentiment classification problem.

This thesis addresses the problem of sentiment classification of documents by filling in these two gaps identified in the field of sentiment classification. The next two sections define the problem tackled in this research and introduce approaches used to solve it.

1.1 Problem definition and learning approaches

First, we define the setting of the problem which will be tackled in this thesis. We assume the availability of several document collections or datasets, which are relatively large and contain comparable numbers of documents. All available datasets except one are labelled, where labels state the sentiment scores. Depending on whether the sentiment classification task is binary or multiclass, sentiments can be either binary (for example, positive or negative) or correspond to document ratings (for example, 1* to 5*). The remaining dataset is unlabelled and, moreover, represents our data of interest which we aim to classify. No labelled documents belonging to the data of interest are available, although the possibility of annotating a small subset, if necessary, is not excluded.

The datasets are assumed to belong to different *domains*. By domain we mean a collection of documents with the same genre (e.g., product reviews) and about the same topic (e.g., electronics). If the word distributions of two document collections are dissimilar enough they are referred to as different domains. This implies that domains can differ by topic, by genre or by both.

The problem described is characterised by the lack or absence of labelled data available from the domain of interest. Such a limited availability of data is a common issue for many natural language processing (NLP) tasks. Fully supervised machine learning techniques usually work best but they need a

substantial amount of labelled data. When a limited amount of labelled data is available, semi-supervised or cross-domain learning approaches are commonly used instead.

Semi-supervised learning relies on a small amount of labelled data and a large number of unlabelled examples from the same domain. It is based on the premise that plentiful unlabelled data can help with learning the model. However, unlabelled data are not always beneficial as the effectiveness of semi-supervised techniques depends partly on the modelling assumptions used (Zhu, 2008). If they are not correct and the amount of labelled data is small, unlabelled data may degrade the performance (Cozman et al., 2003). The most prominent semi-supervised techniques include self-training (Yarowsky, 1995), co-training (Blum and Mitchell, 1998), mixture models with Expectation Maximisation (Nigam et al., 2000), transductive support vector machines (Vapnik, 1998) and graph-based methods (Blum and Chawla, 2001; Zhu and Ghahramani, 2002; Zhu et al., 2003a; Belkin et al., 2006; Talukdar and Crammer, 2009; Subramanya and Bilmes, 2011). The effectiveness of semi-supervised approaches is assessed both by the accuracy achieved and the amount of labelled data used and, thus, the best classifier is the one that has the best trade-off between these values.

Cross-domain approaches (also called domain adaptation) are concerned with the problem of adapting statistical classifiers learned on domains where annotated data are available to new domains. According to the domain adaptation terminology, a domain to which a classifier is adapted, is called

the *target* domain (target domain data are also referred to as *in-domain* data). Domains used for learning the model are called *source* domains (source domain data are also referred to as *out-of-domain* data). Generally, cross-domain approaches consider some labelled and plentiful unlabelled in-domain data; however, in this thesis we assume that no labelled in-domain data are available. Therefore, by domain adaptation we mean *unsupervised* domain adaptation (Jiang, 2008).

The motivation to use cross-domain approaches is based on two premises. On one hand, the acquisition of labelled data tailored for a specific domain is a labour-expensive process, and on the other, there are plentiful labelled data available on the Internet. For instance, in sentiment analysis, a good example is product reviews, whose ratings are already provided by their authors. In contrast, other genres of user-generated content, for example, blogs, forums and social networks, are usually short of labelled data as they have to be annotated manually by human experts. Therefore, a classifier that can be trained on one domain and applied to another may solve the limited labelled data problem. However, supervised techniques are based on the assumption that training and test data are driven from the same underlying probability distribution and, as a result, the straightforward application of machine learning algorithms may lead to poor results if the source and target domains are not alike. The most common ways to tackle the domain adaptation problem are ensembles of classifiers (Aue and Gamon, 2005; Li and Zong, 2008) and various feature transformation algorithms (Blitzer et al., 2006,

2007; Pan et al., 2010, 2011; Glorot et al., 2011). In addition, as cross-domain and semi-supervised techniques share certain similarities, several semi-supervised approaches have been adapted to the cross-domain task, for example, co-training (Yang et al., 2012) and graph-based algorithms (Wu et al., 2009).

In contrast to semi-supervised approaches, cross-domain learning does not require any manual effort, but its results largely depend on the similarity of source and target domains. If there are plentiful labelled data available from other domains, it is crucial to explore what the best strategy would be: the use of existing datasets or the manual annotation of small amounts of target data. Answering this question could help to achieve the highest accuracy using as little human annotation effort as possible.

1.2 Graph-based learning

We adopt graph-based learning as our main method for semi-supervised and cross-domain sentiment classification. This method was chosen for two reasons. First, it can be used in both semi-supervised and cross-domain settings, and second, it can handle multiclass classification. Moreover, graph-based learning has been intensively researched in the last ten years (Zhu et al., 2003a; Joachims, 2003; Talukdar and Crammer, 2009; Subramanya and Bilmes, 2011) and has been proved to be effective for many NLP tasks. In the field of sentiment analysis, graph-based models have been successfully employed for sentiment classification (Pang and Lee, 2004; Goldberg and Zhu,

2006; Wu et al., 2009), automatic building of sentiment lexicons (Hassan and Radev, 2010; Xu et al., 2010), cross-lingual sentiment analysis (Scheible et al., 2010) and social media analysis (Speriosu et al., 2011).

Graph-based algorithms deal with data that can be represented as a weighted graph, with the vertices being data instances and the edge weights corresponding to the similarity between instances. They assume a "manifold structure" of the data, which means that strongly connected instances tend to belong to the same class. If data do not naturally form a graph, the problem of graph construction should be addressed. Graph construction implies finding a similarity function that accurately estimates the similarity between graph vertices and is recognised to be key for the successful performance of graph-based algorithms (Zhu, 2008; Bilmes and Subramanya, 2011).

This thesis explores the most popular and widely used graph-based algorithm, label propagation (LP) (Zhu and Ghahramani, 2002). We also examine and compare several LP modifications which attempt to improve the graph structure used. Although there are two studies which also applied LP to semi-supervised (Goldberg and Zhu, 2006) and cross-domain (Wu et al., 2009) sentiment classification, they do not address some important questions. First, they use certain LP modifications without analysing how different graph structures influence the results. Second, Wu et al. (2009) lacks a study of similarity measures, while Goldberg and Zhu (2006) proposes similarity functions which require additional resources and can only be applied to data from the same domain. Third, Wu et al. (2009) does not suggest any

factors that could impact the results of cross-domain classification. Finally, both studies exploit graph-based learning in certain learning settings and, therefore, there is no comparison between semi-supervised and cross-domain approaches.

1.3 Aims and goals

The aims of the thesis are to address the problem of limited availability of data in sentiment analysis and to advance research in semi-supervised and cross-domain approaches for sentiment classification, considering binary as well as multiclass sentiment scales.

To achieve these aims, the following goals need to be met:

The **first goal** is to investigate various ways of constructing sentiment graphs, which accurately estimate the similarity function and, at the same time, are easy to build, do not require deep linguistic analysis and do not involve manual annotation effort.

The **second goal** is to develop and evaluate a graph-based sentiment analysis system which is able to tackle both binary and multiclass classification and can be used in semi-supervised and cross-domain settings. This implies implementation of several graph-based algorithms and their evaluation in both settings. Evaluation includes establishing optimal parameter values for the algorithms that deliver the best performance, examining the impact of different graph structures and post-processing techniques on the final results to establish the most successful *LP*

modification, studying the sensitivity of the LP modifications to parameter variations, and finally a comparison of graph-based results against the fully supervised classification performance and against state-of-the-art results (where applicable).

The **third goal** is to draw a comparison between semi-supervised and cross-domain approaches to develop recommendations for choosing the most pertinent approach given the data available. In particular, this involves the identification of data characteristics which impact the results of semi-supervised and cross-domain sentiment classification.

The scope of the thesis is limited to the methods and datasets exploited in the study. First, our methodology adopts only graph-based algorithms, which, in turn, are limited to LP (and its modifications) as the most well known and widely used graph-based approach. Second, our data consist of user reviews of different products, which imposes specific style characteristics, such as relatively long texts, a relatively high level of grammatical correctness and a scarcity of Internet slang, emoticons and abbreviations compared, for example, to social network data. The data sizes of all datasets are comparable and the distribution of sentiment classes is similar. Therefore, the effect of these factors on the results of semi-supervised and cross-domain graph-based learning is not studied. As reviews belong to the same genre, we consider domain adaptation only across topics; the possible challenges of domain adaptation across genres are outside the scope of this thesis.

Thus, throughout the thesis, the notions of "domain" and "topic" are interchangeable.

1.4 Structure of the thesis

The thesis comprises eight chapters: introduction, conclusions and six main chapters. The main chapters can be grouped into three parts. The first part, represented by Chapter 2, gives the necessary background information about ongoing research in sentiment classification. The second part, comprising Chapters 3, 4 and 5, describes the initial settings and discusses the methodological aspects of the thesis. Chapters 6 and 7 form the experimental part, where the main contributions of the thesis are presented.

Chapter 2 contains a study of previous work in the field of sentiment classification. All relevant papers are categorised according to four parameters: features, genres, learning approaches and the number of classes. Due to the focus of the thesis, special attention is paid to the studies exploiting semi-supervised and cross-domain approaches.

Chapter 3 introduces the parameters that are essential for the research conducted in the thesis: data, evaluation metrics and baselines. This chapter starts with a description of the data used in our experiments. Then we give an overview of the main preprocessing steps implemented as part of our sentiment classification system, followed by the evaluation metrics used for assessing the experimental results in Chapters 3, 6 and 7. An important part of the chapter is the discussion of feature selection, which aims to establish

the combination of features, yielding the best performance. The best fully supervised classification results are then used as upper bounds for comparison with semi-supervised and cross-domain results. Finally, we provide semi-supervised and cross-domain baselines for the binary and multiclass cases.

Chapter 4 focuses on the core of our sentiment classification system, the graph-based learning approach. First, we provide arguments to support the choice of graph-based learning as our main method. Then we address the problem of graph construction, proposing and evaluating various sentiment similarity measures. The rest of the chapter describes several LP-based inference methods and discusses the post-processing techniques to improve the output values.

Chapter 5 introduces domain complexity and domain similarity, two data characteristics which influence semi-supervised and cross-domain classification results. These characteristics are used in the evaluation conducted in the two subsequent chapters. This chapter also describes a human annotation experiment which estimates the complexity of multiclass sentiment classification for humans and assesses the conformity of review texts with their ratings.

Chapters 6 and 7 presents the evaluation of our graph-based sentiment analysis system in semi-supervised and cross-domain settings. The experiments are carried out for both binary and multiclass classification tasks. Chapter 7 concludes with providing the recommendations which help to choose the best learning setup given the data available.

Chapter 8 revisits the goals established at the beginning of the thesis and presents the original contributions made by this study. To conclude, we define directions for future research.

Chapter 2

Overview of research on sentiment classification

This chapter gives an overview of the research undertaken in the field of sentiment classification. Due to the scope of the thesis, it focuses on semi-supervised and cross-domain studies applied to document-level sentiment classification. Existing research methods and approaches are categorised on the basis of the following four parameters:

- **features** - which reflect the way documents are represented;
- **learning approaches** - which include lexical, supervised, semi-supervised, unsupervised and cross-domain approaches;
- **genres** - which refer to different sources used for testing a method;
- **number of classes** - which include binary and multiclass cases[1].

In Table 2.1, values of the above parameters together with their abbreviations are listed. The most prominent studies on sentiment classification with values of the above parameters used are presented in Table 2.2.

[1]This is done on the basis of the method implementation discussed in the paper and it does not reflect the potentials of the method presented.

Features	
Value	Symbol
Lexical + POS	LxP
Syntactic	Sy
Semantic	Se
Stylistic	St

Approaches	
Value	Symbol
Lexical	L
Supervised	Su
Semi-Supervised	SS
Unsupervised	UnS
Cross-domain	XD

Domains	
Value	Symbol
Reviews	R
Forums&Blogs	FB
Social networks	SN
Other	Ot

Classes	
Value	Symbol
Binary	Bi
Multiclass	Mult

Table 2.1: Values of the parameters with which sentiment classification studies are categorised.

The remainder of the chapter is organised as follows: Section 2.1 reviews common features used for sentiment classification. Section 2.2 gives a brief overview of social media sources which are most exploited in the field of sentiment analysis. Section 2.3 presents important sentiment classification studies grouped according to the learning approach used: lexical, supervised, semi-supervised and cross-domain. Finally, Section 2.4 discusses the existing

Study	Features	Approaches	Genres	Classes
Pang et al. (2002)	LxP	Su	R	Bi
Turney (2002)	Se	L, UnS	R	Mult
Gamon (2004)	LxP,Sy,St	Su	Ot	Bi
Aue and Gamon (2005)	LxP	SS,XD	R	Bi
Pang and Lee (2005)	LxP	Su	R	Mult
Read (2005)	LxP	SS,XD	R,Ot	Bi
Whitelaw et al. (2005)	LxP,Se	Su	R	Bi
Goldberg and Zhu (2006)	LxP	SS	R	Mult
Riloff et al. (2006)	LxP,Sy	Su	R,Ot	Bi
Blitzer et al. (2007)	LxP	XD	R	Bi
McDonald et al. (2007)	LxP	Su	R	Bi
Abbasi et al. (2008)	LxP,St	Su	R,FB	Bi
Dasgupta and Ng (2009)	LxP	SS	R	Bi
Wu et al. (2009)	LxP	XD	R	Bi
Lin and He (2009)	LxP,Se	UnS	R	Bi
Li et al. (2010a)	LxP	SS	R	Bi
Thelwall et al. (2010)	Se	L,SS	SN	Mult
Paltoglou and Thelwall (2010)	LxP	Su	R,FB	Bi
Pan et al. (2010)	LxP	XD	R	Bi
Bai (2011)	LxP	Su	R,Ot	Mult
Bollegala et al. (2011)	LxP	XD	R	Bi
Taboada et al. (2011)	Se	L	R,FB,SN	Bi
He et al. (2011)	LxP,Se	XD	R	Bi
Glorot et al. (2011)	LxP	XD	R	Bi
Paltoglou and Thelwall (2012)	LxP,Se	Su	FB	Mult
Mejova and Srinivasan (2012)	LxP	XD	R,FB,SN	Bi
Li et al. (2012)	LxP,Sy	XD	R	Bi
Zhou et al. (2013)	LxP	SS	R	Bi

Table 2.2: Research studies on sentiment classification.

classification methods which are able to perform multiclass sentiment classification.

2.1 Features

The selection of informative features which best reflect the differences between classes has always been an important issue in classification. This seems harder for sentiment classification due to subtle ways of conveying sentiments, the presence of irony and humour, as well as the risk to misinterpret written language because it is less expressive and more ambiguous than face-to-face interaction. Despite the elusive nature of sentiment, early studies on feature selection showed a reasonably good performance of bag of words (BOW) representations (Pang et al., 2002; Dave et al., 2003). Pang et al. (2002) examined the impact of different lexical features - unigrams, bigrams, adjectives and parts of speech (PoS) - on the results of sentiment classification for movie reviews. They drew the interesting conclusion that simple unigrams with binary weights provide the highest accuracy, outperforming bigrams. In contrast to Pang et al. (2002), Ng et al. (2006) obtained a significant gain in accuracy on the same data by enriching unigram-based BOW representations with bigrams and trigrams. However, they exploited only the most discriminative bigrams and trigrams according to the weighted log-likelihood ratio (Nigam et al., 2000), to avoid high dimensionality problems.

Gamon (2004) analysed three feature sets: n-grams, n-grams + PoS + deep syntactic features (context free phrase structure patterns, transitivity of predicates, tense information, etc.) and PoS + deep syntactic features only.

They concluded that although syntactic features alone led to a substantial drop in accuracy, their contribution is significant when they are used together with n-grams. Riloff et al. (2006) also argued that syntactic information can be beneficial for sentiment analysis as it allows the representation of complex subjective expressions that have non-compositional meanings. The authors merged n-grams and lexico-syntactic patterns on the basis of a specially designed subsumption hierarchy. The idea of the subsumption hierarchy is to identify specific and complex features which bring a substantial gain to the separability of positive and negative classes, and to discard features that perform equal to or worse than more general features subsuming them in the hierarchy. Feature quality was assessed using information gain (IG). The experiments confirmed the effectiveness of both complex features and feature filtering by IG.

Instead of incorporating complex features, Paltoglou and Thelwall (2010) proposed using more accurate weighting functions which they adopted from information retrieval. A Support Vector Machine (SVM) classifier based on the refined feature weights significantly outperformed all known state-of-the-art methods on the movie review dataset.

There are numerous studies that explore the ability of different parts of speech to convey sentiment. Adjectives have been considered good indicators of sentiment since the earliest research on sentiment analysis (Hatzivassiloglou and McKeown, 1997), although they do not usually perform well when used alone (Mullen and Collier, 2004; Whitelaw et al., 2005).

Chesley et al. (2006) analysed verbs in addition to adjectives for the classification of blogs. Their study revealed that positive adjectives are good predictors of neutral and positive blogs, whereas *asserting* and *approving*[2] verbs can be used to identify positive blog posts.

Mullen and Collier (2004) evaluated the importance of semantic information, exploiting Osgood's Theory of Semantic Differentiation (Osgood et al., 1957). The authors expanded a simple vector space model based on lemmas by adding adjectives scored with values for potency, activity and an evaluative factor using the method of Kamps and Marx (2002). Although the hybrid model outperformed simple models based on lemmas and word unigrams, the differences between them were not statistically significant. Another attempt to engage semantic knowledge was made by Whitelaw et al. (2005), who adopted Martin and White's Appraisal Theory (Martin and White, 2005) and semi-automatically constructed a lexicon of adjectives and modifiers with values of attributes from the Appraisal Theory taxonomy. Instead of single adjectives, the authors considered adjectival appraisal groups, which are headed by an appraising adjective and optionally modified by a list of modifiers. They reported a reasonable accuracy for appraisal groups alone, taking into account the low coverage of the constructed lexicon, although the BOW approach significantly outperformed them. However,

[2] These are verb classes in Semantex, which is an automatic text analytics platform for information extraction (Srihari et al., 2003).

appraisal groups gave a substantial improvement in accuracy when used together with the BOW features.

Abbasi et al. (2008) questioned whether stylistic features could give better insights into differences across sentiment classes when analysing web forums. They exploited the following stylistic features: frequency of letters, character n-grams, vocabulary richness measures and frequency of function words, among others. The experiments demonstrated that stylistic features, when complementing lexical and syntactic features, yield a significant gain in accuracy. In general, an increased informality of language inevitably attaches more importance to stylistic features. For example, such features as emoticons, letter repetitions and exclamation mark repetitions are helpful for the sentiment classification of MySpace comments (Thelwall et al., 2010).

Feature reduction is a common technique for many classification tasks as it decreases the dimensionality of a feature space by removing poor or irrelevant features, improving generalisation and diminishing the time needed for training. It has not been thoroughly explored by sentiment analysis researchers, although some existing works indicate its importance for sentiment classification. For example, Gamon (2004) achieved a substantial increase in performance by applying the likelihood ratio (Dunning, 1993) to select the most discriminative attributes from noisy user feedback. Abbasi et al. (2008) demonstrated the effectiveness of genetic algorithms for feature selection, whose combination with IG weights outperformed IG- and SVM-feature selection. In contrast, the results of several widely-exploited machine

learning techniques given in Bai (2011) mostly show a substantial decrease in performance when IG feature reduction is applied.

2.2 Genres

The majority of research in sentiment classification focusses on product reviews. Indeed, they are convenient data for research purposes, as in many environments the reviews are already rated by their authors. As an example, Pang et al. (2002); Mullen and Collier (2004); Pang and Lee (2004); Whitelaw et al. (2005); Riloff et al. (2006); Ng et al. (2006); Abbasi et al. (2008); Paltoglou and Thelwall (2010) and Bai (2011) have all dealt with the same movie review dataset[3]. Their efforts to raise accuracy resulted in an increase from 82.9% to 95.5%.

The social web domain, represented by blogs, forums and social networks is an even more important and necessary source of information due to the rising popularity of social media technologies. Many of these technologies offer a handy interface when accessed from smartphones and other portable devices and are perfect for short notes, sharing ideas and interests, and expressing opinions. Social web data are challenging due to the lack of proper grammar and punctuation, misspellings, made-up words and acronyms. At the same time, they are extremely valuable and timely, as they contain immediate user responses to events and situations, as well as user predictions and speculation about future events. Several studies have

[3]http://www.cs.cornell.edu/people/pabo/movie-review-data/

analysed the correlation between real world events and blog discussions, and found it possible to make accurate predictions, for example, about movie sales (Mishne and Glance, 2006; Sadikov et al., 2009; Asur and Huberman, 2010). Many research studies in sentiment analysis deal with Twitter, the most popular microblogging website (Go et al., 2009; Davidov et al., 2010; Barbosa and Feng, 2010; Kouloumpis et al., 2011; Mejova and Srinivasan, 2012). Another substantial part of research focuses on opinion analysis of news articles and editorials (Devitt and Ahmad, 2007; Wilson et al., 2009; Balahur et al., 2010; Park et al., 2011; Bal, 2014) using, in particular, MPQA (Wiebe and Riloff, 2005; Wilson, 2008) and NTCIR MOAT (Seki et al., 2008) datasets. Other genres include, but are not limited to, legal blogs (Conrad and Schilder, 2007), user feedback (Gamon, 2004) and email (Liu et al., 2003). Thelwall et al. (2012) carried out sentiment strength detection on six social web datasets at once: Youtube, Twitter, a sports forum, MySpace, Digg.com news and BBC forums.

2.3 Approaches

The majority of approaches to sentiment classification fall into one of two groups: lexical or statistical. Lexical approaches are based on either predefined rules or lexicons of sentiment words, which are used to compute the overall sentiment of texts. The lexicons and rules represent expert knowledge about the task and determine the effectiveness of this approach. In contrast, statistical approaches learn a function by matching input texts

and output sentiments from a sample of training data containing both input examples and output values. In this section, the most important representatives of lexical and statistical approaches are described.

2.3.1 Lexical approaches

Lexical approaches are concerned with the use of sentiment lexicons, where each word is assigned to a given sentiment. The final sentiment of a text is normally calculated by applying some scoring function to sentiment-bearing words occurred in the text.

There is a vast amount of research regarding the automatic building of sentiment lexicons. Hatzivassiloglou and McKeown (1997) pioneered this direction, exploiting the idea that conjunctions between adjectives indicate whether they have the same or opposite polarity. In contrast, instead of using syntactic relations, the method of Turney and Littman (2003) was based on semantic association. The authors assumed that words with similar orientations tend to appear together and used Pointwise Mutual Information (PMI) to measure the semantic association of words with a small set of positive and negative seed words. PMI scores were computed using the AltaVista web search engine and semantic association was estimated by the number of pages containing both a given word and a seed word. The method of Turney and Littman (2003) is further referred to as the SO-PMI method. Gamon and Aue (2005) suggested a modified version of this approach, which as well as the co-occurrence hypothesis uses the hypothesis that sentiment

words of opposite orientations tend not to co-occur in the same context. The authors argued that this modification allows a more reliable extraction of sentiment-bearing words, and proved it both qualitatively and quantitatively.

A number of studies exploit WordNet (Miller, 1995), whose explicit semantic relations can help to deduce the polarity of words (Kamps et al., 2004; Kim and Hovy, 2004, 2006; Andreevskaia and Bergler, 2006; Esuli and Sebastiani, 2006a,b; Hassan and Radev, 2010). For example, Kamps et al. (2004) suggested measuring word polarity using *synonymy* relations from WordNet, connecting two words if they occur in the same synset. The polarity of a given word was computed on the basis of the minimal path-length between WordNet synsets of the word and the synsets of two words with opposite meanings: "good" and "bad". Kim and Hovy (2006) also used *synonymy* relations but, in contrast to Kamps et al. (2004), only direct synonyms were considered. Their method requires a relatively large set of predefined positive, negative and neutral words for learning new sentiment words.

In contrast to Kim and Hovy (2006) and Kamps and Marx (2002), who did not take into account the problem of multiple senses, Esuli and Sebastiani (2006a,b) worked at the level of synsets, automatically assigning them three values: level of positivity, level of negativity and level of objectivity. The underlying idea was based on the assumption that synsets with similar orientations tend to have similar glosses. The algorithm required a small set of seed words, representing positive, negative and neutral synsets, which

was further automatically expanded through lexical relations (*synonymy*, *antonymy*, *hypernymy*, etc.). The synsets obtained were then represented as vector models of their glosses and a supervised learning technique trained on the augmented seed set was applied to the remaining WordNet synsets. This method was used to build a well-known linguistic resource, SentiWordNet[4]. Another extension of WordNet was produced by Strapparava and Valitutti (2004), which resulted in the development of WordNet-Affect.

Hassan and Radev (2010) used a Markov random walk model on a graph of related words, where two words were considered related when connected through *synonymy*, *hypernymy* and *similar to* relations. The algorithm was based on the observation that a random walk starting at a given word is likely to reach another word with the same polarity before reaching a word with different polarity. Although the random walk method had a performance comparable to that of the SO-PMI method (Turney and Littman, 2003), it is faster and does not require a large corpus. An interesting and completely unsupervised method was proposed by Zagibalov and Carroll (2008), who searched for negation to identify sentiment-bearing words in Chinese.

Some linguistic sentiment resources were constructed manually: General Inquirer (GI) (Stone et al., 1966), the SO-CAL dictionaries (Taboada et al., 2011), the SentiStrength lexicons (Thelwall et al., 2010) and the MPQA subjectivity lexicon (Wilson et al., 2005), among others. It is worth noting that SentiStrength allows the optimisation of the sentiment strengths of its

[4]http://sentiwordnet.isti.cnr.it/

lexicons using annotated data. This quality is relevant when adapting the sentiment classification engine to different domains.

2.3.2 Supervised classification

Supervised classification requires a corpus of texts labelled with their polarity or sentiment strength. According to numerous studies, when the amount of labelled data is sufficient, this learning approach normally yields the best performance (Pang and Lee, 2008). A pioneering study in supervised sentiment classification (Pang et al., 2002) compared three techniques: naive Bayes (NB), maximum entropy (ME) and SVMs and the results showed a moderate advantage for SVMs over the other methods. In another study, SVMs and ME demonstrated comparable results for the classification of heterogeneous information on the Web (Boiy and Moens, 2009).

Research on supervised methods for sentiment classification has investigated several directions for improving classification results. One popular direction concerns exploring a set of features yielding the best results (Gamon, 2004; Mullen and Collier, 2004; Whitelaw et al., 2005; Kennedy and Inkpen, 2006; Abbasi et al., 2008; Paltoglou and Thelwall, 2010). The main studies in this group were discussed in Section 2.1. Another direction comprises methods which use a combination of classifiers to reduce classification errors given by individual learners (Kennedy and Inkpen, 2006; Prabowo and Thelwall, 2009). For example, Kennedy and Inkpen (2006) combined a lexical approach and SVMs using weighted voting. Their lexical

approach exploited different sentiment lexicons (including GI and a list of positive and negative adjectives (Taboada and Grienve, 2004)), enriched by marking the presence of negatives and intensifiers. Though the lexical approach alone was poor, its combination with SVMs slightly outperformed SVMs on their own.

Similar to Kennedy and Inkpen (2006), Prabowo and Thelwall (2009) hypothesised that the use of multiple classifiers in a hybrid manner can help to improve sentiment classification. However, the classifiers were combined in a sequence rather than in an ensemble. The authors proposed three rule-based methods. The first method used General Inquirer, while the second exploited proper nouns for constructing rules, which were assumed to convey the same sentiment as the whole document. Finally, the last method, called the statistics-based classifier, established a set of rules using sentiment-bearing words automatically rated similar to the SO-PMI method (Turney and Littman, 2003). The hybrid approach, which combines the three rule-based classifiers with SVMs showed a significant advantage over SVMs alone for small datasets. The comparison was not carried out for larger datasets due to a high computational cost of the statistics-based classifier.

A third direction unites papers that represent documents by more fine-grained elements, for example, sentences, and address both coarse- and fine-grained classification problems (Pang and Lee, 2004; McDonald et al., 2007; Zaidan et al., 2007; Li et al., 2010b; Yessenalina et al., 2010; Carrillo de Albornoz et al., 2011). Pang and Lee (2004), when dealing with

movie reviews, hypothesised that objective sentences degrade the sentiment classification of full texts. To filter out objective sentences, a graph min-cut algorithm that takes into account both proximity between sentences in a text and the classification results given by the SVM and NB classifiers was applied. The SVM and NB polarity classifiers trained on filtered texts were compared with those trained on the full documents. As a result, the extraction of subjective sentences was shown to be beneficial only for NB, while SVMs performed marginally better with full documents.

McDonald et al. (2007) assumed that the joint classification of documents and sentences can improve the accuracy of sentiment classification at both levels. The authors suggested an undirected graphical model, where the label of each sentence depends on its neighbouring sentences and the label of the document. In general, inference in undirected graphical models is intractable, but if the document label is fixed the introduced model converts into a chain and the problem can be solved using Viterbi's algorithm. The method slightly outperformed two cascaded classifiers, where one classifies sentences using only a sentence-structured model, and then passes the labels obtained to a document classifier, while the other acts vice-versa.

Zaidan et al. (2007) argued that document annotations enriched with "annotator rationales" can be more effective for sentiment classification than providing a classifier with additional labelled examples. By "annotator rationales", the authors mean the most important words and phrases of a document that indicate its polarity. For each original document, a set of

contrastive examples was constructed by removing one or more annotator rationales. The contrastive examples were used to put additional constraints on the SVM classifier to ensure that the contrastive documents were classified less confidently than the original documents. The results demonstrated a substantial improvement over the baseline SVMs trained only on original documents. Interestingly, training SVMs on annotator rationales only yielded a poor accuracy significantly lower than the baseline. Following the idea of Zaidan et al. (2007), Yessenalina et al. (2010) proposed an unsupervised method for extracting annotator rationales using either OpinionFinder[5] or manually constructed polarity lexicons. Their evaluation showed that automatically constructed rationales are as effective as manually-produced rationales.

The last group of approaches presented here attempt to improve sentiment classification by stepping outside a simple BOW representation (Li et al., 2010b; Bai, 2011). Bai (2011) employed Bayesian networks, which are able to model dependencies among words, proposing an algorithm for learning the Markov Blanket for a sentiment variable. The sentiment variable can have multiple values corresponding to the sentiment expressed in a document. At the first stage, the algorithm establishes a parsimonious vocabulary of words that are expressive enough to capture the overall sentiment of a document. At the second stage, a dependency structure between the words in the vocabulary and sentiment variables is learnt. The experiments showed that

[5]http://mpqa.cs.pitt.edu/opinionfinder/

only several dozen highly predictive words are enough to obtain accurate classification results which are comparable or superior to those of state-of-the-art classifiers trained on BOW representations. Interestingly, the words found important by the algorithm for predicting sentiments are not always sentiment-bearing, for example, "also", "again", "but" and "as", among others. According to the authors, such "results suggest that words that occur often, along with their conditional dependencies and a few strong adjectives, constitute most of the vocabulary needed to express sentiments and perform reasonable predictions" (Bai, 2011, page 741).

Li et al. (2010b) argued that a simple BOW representation is unable to model such complex linguistic phenomena as negation structures, contrast transition, modals and presuppositional structures, which can substantially shift or even invert sentence polarity (Polanyi and Zaenen, 2006). Therefore, the first stage of their algorithm consisted of the automatic detection of sentences with polarity-shifting structures (polarity-shifted sentences). At the second stage three classifiers trained on polarity-shifted sentences, polarity-unshifted sentences and all sentences are trained. Experiments proved the importance of polarity-shifted sentences for correct sentiment classification. Moreover, the final classifier combining three learning models by stacking (Dzeroski and Zenko, 2004) significantly outperformed each of the individual classifiers.

2.3.3 Semi-supervised and unsupervised approaches

Due to the problem of limited data availability, the use of supervised methods is not always possible. Semi-supervised and unsupervised methods attempt to overcome this problem by taking advantage of a plentiful amount of unlabelled data. In this section, several important semi-supervised and unsupervised studies are described.

Dasgupta and Ng (2009) assumed that some texts are easier for sentiment classification to deal with than others and proposed the combination of two techniques to acquire and exploit both easy-to-classify and hard-to-classify data. First, spectral clustering (Ng et al., 2001) was applied to find unambiguous and easy-to-classify reviews. These documents were then used in active learning (Cohn et al., 1994), which attempted to acquire the most ambiguous documents and annotate them manually. Finally, an ensemble of classifiers trained on the same set of ambiguous reviews and different sets of unambiguous reviews was constructed and compared against several baselines. The evaluation verified the effectiveness of each of the suggested steps.

Li et al. (2010a) applied a co-training approach (Blum and Mitchell, 1998) for semi-supervised classification of product reviews. They considered the data from two perspectives: personal and impersonal views. Personal views were defined as those conveyed by personal sentences where the subject is a person, for example, “I am happy with the product”. In turn, impersonal

views were those represented by impersonal sentences where the subject is not a person, for example, "The product is really good". First, a simple heuristic was applied to classify sentences as personal or impersonal, on the basis of which three datasets were composed: reviews with personal sentences, reviews with impersonal sentences and full texts. Then, three ME classifiers trained on these datasets were fused in the co-training procedure. Evaluation showed the significance of each of the proposed steps. First, a random division of sentences yielded a substantial decrease in accuracy compared to the two view division. Second, an ensemble of the three classifiers performed much worse than the co-training approach. Finally, co-training was proved to be better than self-training on the individual classifiers.

Haimovitch et al. (2012) argued that augmenting the amount of unlabelled data can reduce the error rate given by semi-supervised approaches. They conducted large-scale experiments with up to 15 million unlabelled Amazon product reviews employing a bootstrapping approach called AROW (Adaptive Regularisation of Weight vectors) (Crammer et al., 2009). The results demonstrated that the unlabelled data size affects the effectiveness of their approach. For example, increasing unlabelled data from 50K to 1.6M examples reduced the error rate for book reviews by ≈2%. Another interesting outcome of their study is concerned with the amount of labelled data needed for high performance. While increasing the amount of labelled data from 100 to 1000 examples yielded a significant decrease in

the error rate (from 15.2% to 8.4% for book reviews), further growth of the labelled data size did not improve the results much.

Zhou et al. (2013) proposed using deep learning for training semi-supervised sentiment classification models. They introduced a novel approach, called active deep networks (ADN), which combines deep belief networks (Hinton and Salakhutdinov, 2006) with active learning. First, the deep architecture exploiting all unlabelled data and some initial labelled examples is constructed. Then an active learner is applied to identify the most uncertain unlabelled examples and use them for training the networks. To improve the active learning stage by taking into account not only the uncertainty of an example but also the density of the area in which it is found, a modified version of ADN, called information ADN (IADN), was proposed. This helps to choose the most representative examples. The experimental results demonstrated the effectiveness of ADN and IADN compared to previous semi-supervised methods, such as transductive SVMs and the method of Dasgupta and Ng (2009) described above.

An important group of algorithms within the semi-supervised learning approach is graph-based algorithms (Zhu et al., 2003a), which model data as a weighted graph of instances with the edges corresponding to similarity between instances. For document-level sentiment classification, instances are documents and the similarity function reflects the closeness of sentiment between documents. The first attempt to apply graph-based learning to sentiment classification was by Goldberg and Zhu (2006), who proposed

a modified label propagation algorithm (Zhu and Ghahramani, 2002) for multiclass classification of movie reviews. To estimate sentiment similarity between a pair of documents, several similarity measures were tested. The measure that performed best was based on the percentage of positive sentences (PSP) in a document, previously introduced by Pang and Lee (2005). For computing PSP scores, review sentences were classified as either positive or negative using a binary classifier trained on an external "snippet" dataset. The snippet dataset comprised 10 662 short texts taken from the movie reviews on rottentomatoes.com, where the ratings for snippets were assigned on the basis of the ratings of their original reviews. Using the PSP scores, each document was represented as a vector (PSP, 1-PSP) and the similarity between two documents was measured as the cosine similarity between the corresponding vectors. For relatively small amounts of labelled data (less than 200 documents), the graph-based results demonstrated a considerable improvement over the accuracies given by supervised SVM regression. In contrast, for larger amounts of labelled data, supervised SVM regression generally performed better. The method of Goldberg and Zhu (2006) is implemented as part of our graph-based sentiment analysis system and, therefore, a more detailed description of the algorithm can be found in Chapter 4.

Turney (2002) was the first to tackle the sentiment classification problem in an unsupervised manner. He assumed that lexical association of two words and their similarity are related, i.e., words with similar orientation

tend to co-occur. Instead of considering isolated words, he extracted two-word phrases which contain adjectives and adverbs and satisfy a set of specific linguistic patterns, for example, JJ NN or RB JJ. Phrases were preferred to isolated words for introducing some context which could help to disambiguate domain-dependent and context-dependent sentiment words, for example, "unpredictable plot" and "unpredictable steering". The semantic orientation of phrases was measured using the SO-PMI method as explained in Section 2.3.1. The sentiment of documents was computed by averaging the semantic orientations of their phrases. This approach gave reasonable results taking into consideration its being completely unsupervised.

Read and Carroll (2009) extended the SO-PMI method by exploring three types of similarity measures: lexical association measures as in Turney and Littman (2003) and two second-order similarity measures - semantic spaces and distributional similarity. The overall sentiment of a document was computed on the basis of the sentiments of its features, which in turn were represented as a sum of similarity scores between a feature and a set of predefined prototypical words. Seven positive and seven negative words were selected as polarity prototypes. The evaluation of the proposed word similarity method showed that its performance is independent of domains, topics and time-periods. In addition, a comparison with supervised techniques suggested that the word similarity method can be more beneficial than supervised techniques when the task involves multi-domain datasets.

Read (2005) proposed the acquisition of training data in an unsupervised

manner which exploits the idea that emoticons and their contexts convey similar sentiments. The author collected data from Usenet newsgroups and extracted pieces of text close to emoticons using different context windows. Unfortunately, the results were not very good: the best classifier trained on 20 000 articles could only achieve 70.1% accuracy. The author explained the low accuracy by the high level of noise present in the automatically acquired labelled data. This unsupervised labelled data acquisition approach was adopted in several subsequent studies (Go et al., 2009; Pak and Paroubek, 2010) for sentiment analysis on Twitter.

Another unsupervised method developed by Zagibalov (2010) exploits a small number of sentiment-bearing seed words and a bootstrapping strategy. First, all documents are classified according to the seed words and all lexical units are weighted on the basis of their frequency in positive and negative documents. Then, the procedure is repeated, which leads to new document labels and updated weights of lexical units. This process iterates until convergence is achieved.

Lin and He (2009) proposed a joint sentiment-topic (JST) model as another unsupervised approach to sentiment classification. Their model extends Latent Dirichlet Allocation (LDA) (Blei et al., 2003) by adding a sentiment layer, which simultaneously allows the extraction of mixture of topics and the detection of their sentiments. To refine the model, some prior knowledge about sentiment-bearing and opinionated words was incorporated, which in fact added some supervision to the method. The authors used the

MPQA subjectivity lexicon together with the removal of objective sentences in a supervised manner similar to Pang and Lee (2004). The results obtained on the movie review dataset were lower than those of fully supervised approaches but are still surprisingly high when we consider the almost unsupervised nature of the method. There are also a number of similar studies where topic models are exploited for extracting aspects of reviews (Mei et al., 2007; Titov and McDonald, 2008; Brody and Elhadad, 2010; Zhao et al., 2010) although their main objective was to produce summaries rather than to detect the sentiment of a document.

2.3.4 Cross-domain learning

Cross-domain learning is another way to address the limited availability of labelled data and is concerned with adapting statistical classifiers trained on source domains to a target domain. It can be addressed either in semi-supervised or in unsupervised settings, where the former exploit a small subset of labelled data from the target domain, while the latter do not require any labelled target data. As this thesis tackles the unsupervised cross-domain problem, this section mostly covers unsupervised domain adaptation studies.

Early work on domain adaptation employs ensembles of classifiers trained on different source domains (Aue and Gamon, 2005; Li and Zong, 2008). For example, Aue and Gamon (2005) studied several possibilities for combining data from domains with known annotations and concluded that an ensemble

of classifiers in a meta-classifier gives a higher performance than a simple merging of all features.

Another group of studies unites the algorithms which attempt different transformations of the feature space to map source and target domain features (Blitzer et al., 2007; Pan et al., 2010; Bollegala et al., 2011; Li et al., 2012). The pioneering work here is structural correspondence learning (SCL) (Blitzer et al., 2006, 2007). Its underlying idea is to find correspondences between features from source and target domains through the modelling of their correlations with pivot features. Pivot features are features occurring frequently in both domains and, at the same time, serving as good predictors of document classes, such as the general sentiment words "excellent" and "awful". The extraction of pivot features was based on their frequency in source and target corpora and their mutual information with positive and negative source labels. The correlations between the pivot features and all other features were modelled using supervised learning of linear pivot predictors to predict occurrences of each pivot in both domains. The proposed method was tested on review data from four domains (books, DVDs, kitchen appliances and electronics) and demonstrated a significant gain in accuracy for most domain pairs compared to a baseline cross-domain classifier. However, for a few domains the performance degraded due to feature misalignment: the narrowness of the source domain and diversity of the target domain created false projections of features in the target domain.

The authors proposed correcting this misalignment with a small amount of annotated in-domain data.

Spectral feature alignment (SFA), introduced by Pan et al. (2010), advocates the same idea as SCL, i.e., an alignment of source and target features through their co-occurrences with general sentiment words. But instead of learning representations of pivots in source and target domains, the authors used spectral clustering to align domain-specific and domain-independent words into a set of feature-clusters. The clusters were then used for the representation of all data examples and for training the sentiment classifier. This new solution yields a significant improvement in cross-domain accuracy compared with SCL for almost all domain pairs.

The method suggested by Bollegala et al. (2011) also uses word co-occurrences. However, unlike in previous methods, the adaptation from multiple source domains was carried out. It consists of the automatic construction of a sentiment-sensitive thesaurus where each lexical element (unigram or bigram) is connected to a list of related lexical elements which most frequently appear in the context expressing the same sentiment. This thesaurus is then used in the training step to expand document features with related elements to overcome the feature mismatch problem. The accuracy obtained outperformed the accuracies given by SCL and SFA averaged over source-target domain pairs with the same target domain.

Li et al. (2012) also exploit general sentiment words as a bridge to find correspondences between source and target features. However, instead of

using co-occurrences, their algorithm is based on the co-extraction of topic and sentiment words using syntactic patterns. First, highly confident seeds of sentiment words (commonly used in both domains) and topic words are extracted. Next, the seeds are expanded using relational adaptive bootstrapping, which extracts the most confident topic and sentiment words based on two cross-domain classifiers and then prunes them on the basis of the syntactic patterns identified during previous iterations. The SVM classifier trained on the sentiment lexicons demonstrated a small but statistically significant improvement over baseline SVMs trained on all unigrams and bigrams.

Glorot et al. (2011) argued that deep learning representations constructed on the basis of both source and target datasets can yield better transfer across domains. Their approach consists of two steps. First, a hierarchy of features is learned on the basis of all the data available in an unsupervised manner using a Stacked Denoising Auto-encoder (Vincent et al., 2008). Second, the new representation of the source data is used for training a linear SVM classifier to be applied to the target data. The approach was evaluated on the multi-domain sentiment dataset and showed a substantial improvement over previously published cross-domain methods such as SCL and SFA.

He et al. (2011) explored the effectiveness of topic models for domain adaptation by applying the joint sentiment-topic modelling (Lin and He, 2009) reviewed above to a merged corpus of source and target datasets. The original source documents were then augmented with the sentiment-

bearing topics obtained and were used for training a supervised classifier. The evaluation on target documents augmented with the sentiment-bearing topics showed good results which outperform SCL and are comparable to the accuracies delivered by SFA.

Several studies employed graph-based learning for cross-domain sentiment classification (Tan et al., 2007; Wu et al., 2009). Wu et al. (2009) proposed a graph-ranking algorithm for the binary classification of Chinese user reviews and demonstrated its competitiveness with SCL. The graph-ranking algorithm is implemented as part of our graph-based sentiment analysis system and is addressed in more detail in Chapter 4. Although it was introduced by Wu et al. (2009) as a novel approach we reveal its strong similarity with LP.

All previous studies assume that both source and target domains belong to the same genre. However, some social media sources, such as product reviews, are rich in labelled data, whereas others, such as tweets, are generally scarce in labelled data. This suggests that the possibility of performing accurate domain adaptation across genres could solve the problem of the limited availability of labelled data. The first attempt to explore cross-genre sentiment classification was by Mejova and Srinivasan (2012), who experimented with three social media sources: reviews, blogs and microblogs (Twitter), on five different topics: movies, music albums, smart phones, computer games, and restaurants. The evaluation showed very promising results, indicating that classifiers built on reviews are the most generalisable

to other genres and in many cases deliver performance comparable to that of an in-domain classifier. Twitter was also found to be a good source of training data as the Twitter-based model performed reasonably well, especially for blogs.

2.4 Remarks on multiclass classification

Most of the studies from Table 2.2 address the binary classification problem. Multiclass sentiment classification (also referred to as the rating-inference problem) is naturally more difficult and not all machine learning methods are designed to undertake classification of more than two classes. The most common ways to tackle multiclass problems are to apply regression or classification methods in a one-vs.-all or one-vs.-one fashion (Pang and Lee, 2005; Paltoglou and Thelwall, 2012). Pang and Lee (2005) tested three techniques for the multiclass classification of movie reviews using 3-point and 4-point scales. In addition to SVM regression (SVR) and one-vs.-all SVMs (OVA SVMs), a metric labelling method was applied. This is a supervised analogue of the graph-based algorithm, presented in Goldberg and Zhu (2006). Metric labelling incorporates classification results given by supervised learning, as well as the similarity between unlabelled examples and their nearest labelled neighbours. The method exploits an additional parameter which controls a trade-off between the impacts of the supervised solution and the rating given by nearest neighbours of a document. Evaluation proved the importance of the information provided by document

neighbours: in most cases the results from SVR and OVA SVMs were improved when the similarity between documents was taken into account.

Graph-based learning represents an efficient and effective solution for multiclass classification. As shown in this chapter, it has successfully been applied to semi-supervised and cross-domain tasks (Goldberg and Zhu, 2006; Wu et al., 2009). In addition, the study of Pang and Lee (2005) demonstrated the benefit of graph structures for supervised classification. Another advantage of graph-based algorithms over, for example, OVA SVMs, is their ability to deal with many classes without substantial additional computational costs.

2.5 Summary

This chapter presented a detailed review of research in the field of sentiment classification, with an emphasis on document-level classification. All studies were categorised and compared on the basis of four parameters, features, domains, learning approaches and the number of classes, and each section of this chapter addressed the research undertaken with respect to a corresponding parameter. In Section 2.1, the most common features used for sentiment classification and their effect on classification results were described. Section 2.2 listed popular social media sources in the field of sentiment analysis. Section 2.3 introduced the most notable representatives of the main learning approaches. Finally, in Section 2.4, some existing multiclass studies were briefly described.

Chapter 3

Data, preprocessing and baselines

Data determines the outcomes of classification methods. In sentiment classification, many different genres are usually dealt with: product reviews, blog posts, tweets and other social network data, amongst others (Chapter 2). In this thesis we experiment with product reviews belonging to the multi-domain sentiment dataset[1]. We chose this data for two reasons. First, it comprises reviews on many topics, where each topic is represented by a relatively large amount of data. Thus, it is possible to use the data in our cross-domain experiments. Second, the datasets are used by many studies on cross-domain and semi-supervised sentiment classification, which makes comparison of our results with those of state-of-the-art research straightforward.

The chapter is organised as follows. Section 3.1 describes our data. Section 3.2 reports the main preprocessing steps. Section 3.3 presents the evaluation metrics, which are used in the thesis for assessing and comparing results. Section 3.4 describes our study on feature selection where different feature sets and feature reduction techniques are tested to establish those giving the highest performance. The best results are then exploited as

[1] http://www.cs.jhu.edu/∼mdredze/datasets/sentiment/

the performance upper bounds in the semi-supervised and cross-domain experiments. Finally, in Section 3.5, semi-supervised and cross-domain baselines are given which are used later as the lower bounds of performance.

3.1 Data

Our data comprises Amazon product reviews from the multi-domain sentiment dataset. This dataset contains reviews on 25 product types. However, some of the product types either have too little data, such as musical instruments and office products, or are highly unbalanced, like gourmet food and jewellery & watches. Thus, we selected only seven topics with 2000 reviews each and equal numbers of positive and negative documents:

- Books (BO): book reviews;
- Electronics (EL): reviews of electronic devices and accessories;
- Kitchen (KI): kitchen appliances and housewares reviews;
- DVDs (DV): reviews of films, series, documentaries on DVDs;
- Music (MU): audio CD and DVD reviews;
- Toys (TO): reviews of games, educational toys, arts & crafts, etc;
- Health (HE): health and personal care product reviews.

The data from the first four topics (BO, EL, KI and DV) has been intensively exploited in sentiment analysis and have become a gold standard used to compare different classification techniques (Blitzer et al., 2007; Pan et al., 2010; Dasgupta and Ng, 2009; Li et al., 2010a). The last three topics (MU, HE and TO) were selected randomly from the product types encompassing 2000 reviews. We use two different data configurations: binary and multiclass data. Reviews in the multiclass data preserve the initial ratings, 1*,2*,4* and 5*, a total of four sentiment classes. The distribution of classes is shown in Figure 3.1. To obtain the binary datasets, 1* and 2* reviews were considered to be negative and 4* and 5* reviews were considered to be positive. As previously mentioned, the binary datasets are balanced: they contain 1000 positive and 1000 negative reviews within each domain.

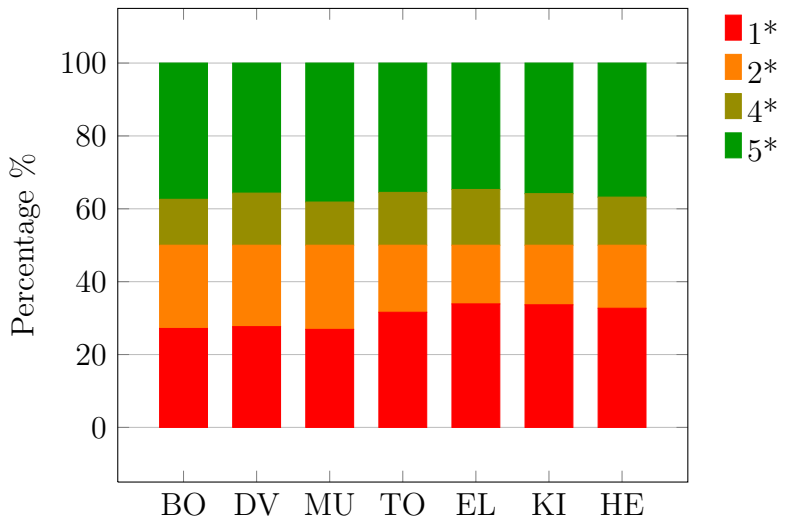

Figure 3.1: Class distribution for the multiclass datasets.

Table 3.1 provides some statistics about the data: the number of tokens (all words in the corpus), the mean number of tokens per document, the number of types (all the individual, different words in the corpus) or vocabulary size, the type/token ratio (TTR) and the percentage of rare words (i.e. words that appear in less than 10 documents in a corpus). TTR is a standard measure of vocabulary richness used in corpus linguistics (Biber et al., 2002)[2]. The percentage of rare words can also indicate vocabulary richness if the corpus is big enough. Table 3.1 gives some insights into the data. In particular, there are substantial differences between BO, DV and MU, on one side, and EL, KI and HE, on the other. BO, DV and MU have longer reviews, a higher TTR and a higher percentage of rare words, which implies greater vocabulary richness. The MU domain is an extreme case in this respect as it has the highest values for both vocabulary richness measures. This conforms to our intuition that reviews of books, movies and music are in general more sophisticated and diverse because of their cultural component.

3.2 Preprocessing

As part of the thesis, a Java-based sentiment analysis system was developed. It consists of a preprocessing module and a classification module. The major part of the preprocessing module exploits GATE[3], an open source

[2]Strictly speaking this measure depends on the data size (Baayen, 2001) but in our case, when the difference between data sizes is relatively small, we can ignore this factor

[3]http://gate.ac.uk/

corpus	# tokens	mean # tokens/doc	# types	type/token ratio	% of rare words
BO	367k	183.4	27k	0.0725	89.5%
DV	401k	200.4	28k	0.0703	89.3%
MU	303k	151.5	23k	0.0748	90.3%
TO	208k	104.0	13k	0.0626	87.8%
EL	239k	119.3	14k	0.0583	86.2%
KI	200k	100.0	12k	0.0608	86.4%
HE	190k	95.1	12k	0.0641	87.2%

Table 3.1: Review corpora statistics.

tool for text processing. In particular, we used the ANNIE plugin[4] for the main preprocessing steps, such as tokenisation, sentence splitting and POS tagging. Other steps include conversion to lower case, substitution of verbal short forms by their full forms, punctuation removal and number replacement (Figure 3.2). To examine which word form is the most beneficial for sentiment classification, we performed stemming with the Snowball stemmer[5] and lemmatisation with the GATE Morphological Analyzer[6].

Due to the importance of negation for the results of sentiment classification, we implemented a simple rule-based approach which deals with several negating words: "not", "no", "nothing" and "never". The rules for treating each of these negating words are slightly different due to the idiosyncrasies of their use in English. However, our main assumption is similar for all cases that negating words usually negate subsequent verbs, adjectives, nouns and adverbs belonging to the same clause. For example,

[4]http://gate.ac.uk/gate/doc/plugins.html#ANNIE
[5]http://gate.ac.uk/sale/tao/splitch21.html#sec:parsers:stemmer
[6]http://gate.ac.uk/sale/tao/splitch21.html#sec:parsers:morpher

"The action was NOT anything **new**.", "It has really easy refill - NO **mess**, NO **fuss**.", "Everything was telegraphed and NOTHING was **original**." and "I would NEVER **recommend** this product to anyone.".

We elaborated a set of rules which find words affected by a negating word and append the tag "NOT" to them. The search for the words affected by negation is run in a window of 4 words[7] and terminates when a punctuation mark or conjunction is met. For the negating word "no" the search ends after the first noun. As a result of this procedure, the previous examples transform to "The action was anything **new_NOT**.", "It has really easy refill - **mess_NOT**, **fuss_NOT**.", "Everything was telegraphed and was **original_NOT**." and "I would **recommend_NOT** this product to anyone.". Therefore, the features "new" and "new_NOT", "mess" and "mess_NOT", "recommend" and "recommend_NOT", etc. are treated

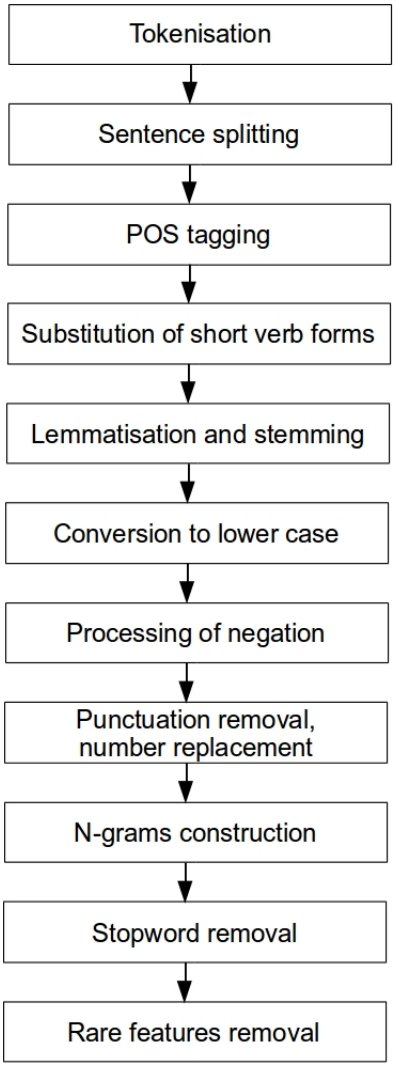

Figure 3.2: Preprocessing steps

[7]This number was established empirically

differently by the machine learning algorithm. Of course, this simple approach cannot cope with more complex cases, for example, a conjunction of two adjectives: "It's not interesting and relevant" or with sentences where other negating words are involved, for example, "It is neither interesting nor relevant". We assume that these situations are rare enough that we do not need to take them into account.

The pioneering work of Pang et al. (2002) on statistical sentiment classification demonstrated the higher effectiveness of unigrams compared to unigrams+bigrams. However, several later studies showed the opposite (Dave et al., 2003; Andreevskaia and Bergler, 2008). Naturally, bigrams can capture some contextual information and give features a higher discriminative power. In the examples "The book is nothing more than mediocre at best" and "This is the best book I have ever read", the word "best" conveys both positive and negative sentiments, yet, its negative meaning in the first sentence only becomes clear when it is considered with the preposition "at". To examine the impact of n-gram size on sentiment classification, we implemented the "n-grams construction" module, which allows us to experiment with n-grams of different sizes.

The stopword removal step eliminates stopwords: commonly used function words that do not have meaning on their own. The list of stopwords, downloaded from textfixer.com[8], was adapted to sentiment classification by removing from it modal verbs ("could", "would", "might") and some adverbs

[8]http://www.textfixer.com/resources/common-english-words.txt.

("again", "too") due to their strong subjective connotations. Experiments with and without stopword removal show similar results when all features are used. Finally, we applied the conventional procedure of excluding rare features from the space vector model with a threshold of 5 (features that occur less than 5 times in a corpus are removed). We refer to this reduced feature set as the full feature set.

3.3 Evaluation metrics

In this section, we provide an overview of common metrics used to evaluate text classification results. We conclude with the choice of measures that suit our task best and are consistent with related work.

Accuracy, defined as the proportion of correctly classified documents to the total number of documents, is the simplest classification measure. It computes the overall effectiveness of a classifier but does not give an understanding of its performance on individual classes. To obtain deeper insights into classification results, one can use confusion tables, recall and precision (van Rijsbergen, 1975), which evaluate the effectiveness of the classifier on each class. Confusion tables are an easy way to illustrate classification results regarding one of the classes (see Table 3.2).

	True C_k	True $\neg C_k$
Predicted C_k	true positive (tp_k)	false positive (fp_k)
Predicted $\neg C_k$	false negative (fn_k)	true negative (tn_k)

Table 3.2: Confusion table for the class C_k

Using the notation from Table 3.2, recall, $R(C_k)$, and precision, $P(C_k)$, can be computed as:

$$R(C_k) = \frac{tp_k}{tp_k + fn_k}, \quad P(C_k) = \frac{tp_k}{tp_k + fp_k} \tag{3.1}$$

$R(C_k)$ refers to the proportion of correctly identified documents for the class C_k out of the total number of documents in this class. In turn, $P(C_k)$ is the proportion of correctly identified documents for the class C_k out of the total number of documents assigned to this class. These measures are especially useful for unbalanced datasets when the least frequent classes tend to be misclassified as members of more numerous classes. The result of such misclassifications is usually low recall for infrequent classes.

F-score (also called F-measure) is the harmonic mean of precision and recall and is used when a trade-off between these two measures is needed:

$$F_\beta(C_k) = \frac{(1 + \beta^2)P(C_k)R(C_k)}{\beta^2 P(C_k) + R(C_k)} \tag{3.2}$$

Parameter β in (3.2) gives different weights to either precision or recall depending on the classification task. Most commonly, precision and recall are considered equally important and, thus, $\beta = 1$.

To find the average effectiveness of a classifier on the basis of its performance on individual categories, one can apply microaveraging and macroaveraging methods (Manning et al., 2008). Microaveraging estimates recall, precision and F-score over the whole collection rather than for

individual classes and tends to reflect the performance given by frequent categories. In sentiment classification when each document is assigned a single category, microaveraged precision, recall and F-score are equal and coincide with classification accuracy. Macroaveraging computes the average recall, precision and F-score over classes, which equalises the impact of each class independent of their size:

$$\begin{aligned} macroR &= \frac{1}{m}\sum_{k=1}^{m} R(C_k), \\ macroP &= \frac{1}{m}\sum_{k=1}^{m} P(C_k), \\ macroF_1 &= \frac{1}{m}\sum_{k=1}^{m} F_1(C_k). \end{aligned} \tag{3.3}$$

where $C_1, ..., C_m$ are classification categories and m is the number of categories.

Mean squared error (MSE) is another popular evaluation metric used for assessing regression results. This measure is especially important for multiclass sentiment classification, which can be seen as a sentiment regression problem with sentiment values given by a continuous rating function. MSE has been exploited in various sentiment analysis studies for evaluating multiclass results (Wilson, 2008; Paltoglou and Thelwall, 2012). Unlike accuracy, it penalises results in accordance with their distance from the actual sentiment. However, MSE does not reflect the number of exact matches between predicted and actual sentiments and, thus, should be used as a complementary measure to classification metrics.

In the thesis, we use three evaluation metrics:

1. **Accuracy**. We use accuracy (equal to microaveraged F-score, $microF_1$) as we are interested in the correct classification of as many documents as possible. Moreover, it eases the comparison with the state-of-the-art results on the binary data where mostly accuracies are reported.

2. **Macroaveraged F-score**. The multiclass data is not balanced and, therefore, it is important to make sure that not all examples are assigned only to "very positive" and "very negative" categories due to their largest size, as then the multiclass classification will degenerate to the binary case. Macroaveraged F-score, $macroF_1$ helps to avoid this as its value can drop if one of the classes is represented poorly in the final results.

3. **The mean of macro- and microaveraged F-score**. To select the best results that maximise both microaveraged and macroaveraged F-scores, a mean F-score, $\bar{F}_1$, is introduced, which averages accuracy and $macroF_1$:

$$\bar{F}_1 = \frac{microF_1 + macroF_1}{2} \tag{3.4}$$

4. **MSE**. We also report the MSE values to ensure a small variance between actual and predicted sentiments.

To compare the performance obtained by different classification techniques a paired t-test with significance level $\alpha = 0.05$ is used.

3.4 Feature selection

In this section, we describe our study of the features that give the highest performance across all domains. Our objective is to identify the best features in the following dimensions: word forms (tokens vs. stems vs. lemmas); n-gram sizes (unigrams vs. unigrams + bigrams); feature weights (binary weights vs. frequency vs. tf-idf vs. idf vs. Delta tf-idf vs. Delta idf) (Martineau and Finin, 2009; Paltoglou and Thelwall, 2010). In addition, we aim to reduce the feature set size by taking into consideration feature informativeness or discriminative power. In this respect, we examine which feature reduction technique is the most beneficial for sentiment classification and what number of features provides the highest accuracy. We also assess whether feature reduction is capable of improving the performance given by the full feature set. Since the results may be different for binary and multiclass datasets, we conduct our analysis separately for each case.

All experiments are carried out using an SVM classifier with a 5-fold cross-validation setup. The choice of this learning algorithm is based on numerous experimental confirmations of its effectiveness for sentiment classification (Pang and Lee, 2008). We use the LIBSVM library (Chang and Lin, 2011) and a linear kernel function to train the classifier.

3.4.1 Related research

An overview of research concerning with the choice of good features for sentiment classification was given in Section 2.1. Here we point out some contradicting conclusions obtained by different studies, which make it necessary to carry out new investigations. For example, as mentioned above, previous work did not agree about which n-gram size gives the best performance. It is also not clear whether the weighting of features is able to improve accuracy. On one hand, a simple representation, i.e. binary weights (Pang and Lee, 2008), has been shown to be effective, but, on the other hand, several studies have demonstrated that a significant accuracy gain can be obtained by using complex feature weighting schemes (Paltoglou and Thelwall, 2010; Kim et al., 2009).

Feature reduction (also called feature selection) is an important preprocessing step in the text classification task that aims to optimise the performance of a classifier by reducing feature space dimensionality (Guyon and Elisseeff, 2003). When the number of features is many times higher than the number of training examples, the learning model might become very complex and overfit the data. Although SVMs are highly resistant to overfitting through the regularisation parameter C (Cortes and Vapnik, 1995), reducing the number of features makes large problems computationally efficient. Moreover, many studies have shown that correctly chosen features can substantially improve classification accuracy (Gamon,

2004; Abbasi et al., 2008; Yang and Pedersen, 1997; Mladenic and Grobelnik, 1999). Basic feature reduction includes standard filtering of rare words using a given threshold. Yet not all frequent features are discriminative and serve as a good indicator for a class. For example, the word "book", though quite frequent in book reviews, is equally probable in both positive and negative contexts. Therefore, this feature does not add any relevant information to help separate positive reviews from negative ones and can be easily discarded.

All feature reduction techniques aim to find the subset of discriminative features that provides the best performance. There are two main approaches to feature reduction: filtering and wrappers. Filtering chooses relevant features by ranking them according to some measure of goodness, and wrappers use a classifier as a black box to induce the feature subset delivering the highest performance (John et al., 1994; Guyon and Elisseeff, 2003). Filters are simple and fast but their solution is usually suboptimal since they consider features independently of each other and allow redundant features. Wrappers, though providing an optimal subset for a given learning algorithm, involve searching the space for all possible feature subsets which could be computationally expensive for a high-dimensional feature space. For reasons of speed, simplicity and also popularity among researchers (Forman, 2003; Gamon, 2004; Riloff et al., 2006), we adopt the filtering approach as our feature reduction technique. There are numerous functions that can be used for feature ranking, such as Information Gain (IG), χ^2, Mutual Information, Odds Ratio (OR), etc. Previous studies do not agree on the metric that

performs best for all datasets and machine learning algorithms. For example, Yang and Pedersen (1997) revealed that IG, χ^2 and document frequency are strongly correlated and are equally effective even when very aggressive feature removal (over 90%) is applied. In contrast, Mladenic and Grobelnik (1999) reported low performance for IG and instead suggested OR and its variants, as they performed best on their dataset. Forman (2003) confirmed the advantage of IG over all previously studied metrics, however, it did not outperform a new measure called Bi-Normal Separation, proposed in their paper. Concerning work on sentiment classification, Gamon (2004) used a variant of OR - likelihood ratio (LR) (Dunning, 1993) - for the sentiment classification of noisy user feedback, which yielded a substantial gain in performance. Ng et al. (2006) successfully exploited a similar metric called Weighted Log-Likelihood Ratio (WLLR) (Nigam et al., 2000) for sentiment classification of movie reviews.

On the basis of previous research, we chose IG, LR and WLLR to perform feature reduction. Let us introduce their formal definitions. If $\{C_i\}_{i=1}^m$ is a set of sentiment classes, then the scores of a feature f with respect to these metrics are given by formulas 3.5-3.7:

$$IG(f) = -\sum_{i=1}^{m} P(C_i) \log P(C_i) + P(f) \sum_{i=1}^{m} P(C_i|f) \log P(C_i|f) + P(\bar{f}) \sum_{i=1}^{m} P(C_i|\bar{f}) \log P(C_i|\bar{f}) \quad (3.5)$$

$$LR(f) = \max_{C_i} \frac{P(f|C_i)}{P(f|\bar{C}_i)} \quad (3.6)$$

$$WLLR(f) = \max_{C_i} P(f|C_i) \log \frac{P(f|C_i)}{P(f|\bar{C}_i)} \tag{3.7}$$

where $\bar{f}$ is the absence of feature f and $\bar{C}_i$ is non-membership in C_i.

LR prioritises features with the highest discriminative power, i.e., those occurring many times more frequently in one of the classes. However, it does not take into account the feature frequency in the data, which can lead to a high ranking of rare and quite specific features. WLLR overcomes this drawback by multiplying feature discriminative power by feature frequency. The preference of LR for rare but discriminative features can be observed in Table 3.3. Both IG and WLLR rank the feature "money" higher than the more specific but more likely to be found in negative contexts "your money", while LR does the opposite. Unlike IG and WLLR, LR does not give a high rank to "best" which can be both positive and negative, but prefers a more precise and definitely negative "at best". WLLR has another shortcoming: due to its dependence on feature frequency it can give a high preference for topic words. For instance, it ranks the words "rock" and "music" 27th and 36th respectively, while they receive much lower scores from IG (53rd and 303rd) and LR (942nd and 2055th). In contrast to LR and WLLR, IG ranks features on the basis of their informativeness, i.e. the amount of information obtained for a class prediction by knowing a feature's presence or absence in texts. This implies that IG does not prioritise either topic terms or rare words. In fact, rare words are not informative due to the low probability of

their occurrence and topic terms have high entropy because they are equally possible in positive and negative contexts.

Rank	IG	LR	WLLR
1	great	dull	great
2	boring	well worth	best
3	worst	awful	NUM of
4	bad	mess	boring
5	money	your money	bad
6	favorite	waste NOT	worst
7	best	highly recommended	favorite
8	dull	great album	money
9	NUM of	holds	like
10	awful	disappointing	love
11	sounds like	worst	dull
12	your money	boring	sounds like
13	wonderful	terrible	awful
14	highly	favorite	wonderful
15	favorites	will love	your money
16	terrible	mood	highly
17	even NOT	at best	also
18	amazing	favorites	well
19	highly recommended	joke	amazing
20	mood	poor	even NOT

Table 3.3: 20 most discriminative features from the music domain according to three feature ranking functions: IG, LR and WLLR.

3.4.2 Binary classification

First, we ran a series of experiments to establish the optimal n-gram size. Our observations are concordant with Dave et al. (2003): the SVM classifier gives significantly better performance on our data when bigrams are also included in the feature set. Figure 3.3 presents in-domain accuracies for all

seven domains when features are either unigrams or unigrams+bigrams. We used tokens as word forms and binary weights as feature weights although a similar picture is observed for other word forms and feature weights.

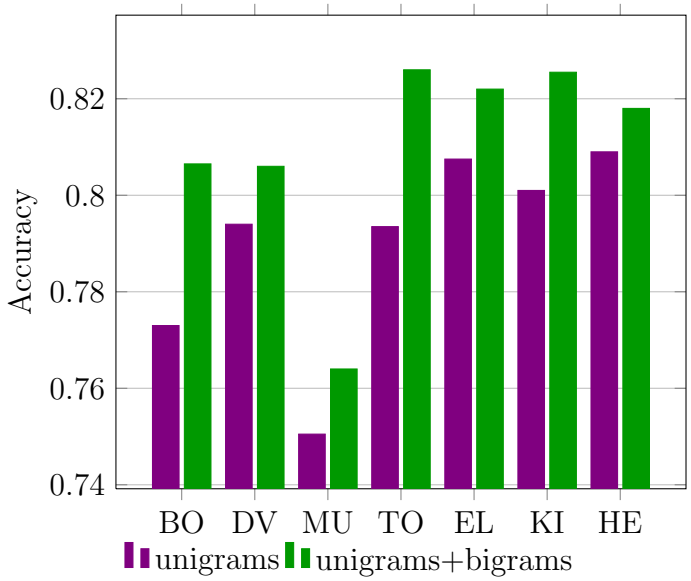

Figure 3.3: In-domain accuracies for the unigram and unigram+bigram vector model (binary case).

Second, we experimented with different feature weighting schemes. Unlike Paltoglou and Thelwall (2010) and Kim et al. (2009), we concluded that more complex feature weights like Delta idf and Delta tf-idf do not improve performance compared to their simpler analogues. We also observed that binary and idf weights outperform frequency and tf-idf weights in most cases (Figure 3.4). For some domains, such as KI, DV, TO and HE, the difference is small, but it is statistically significant for BO, EL and MU. Interestingly, there is little difference in the results within the group of weights based on

frequency (frequency, tf-idf, Delta tf-idf weights) as well as within the group of weights based on word presence (binary, idf and Delta idf weights).

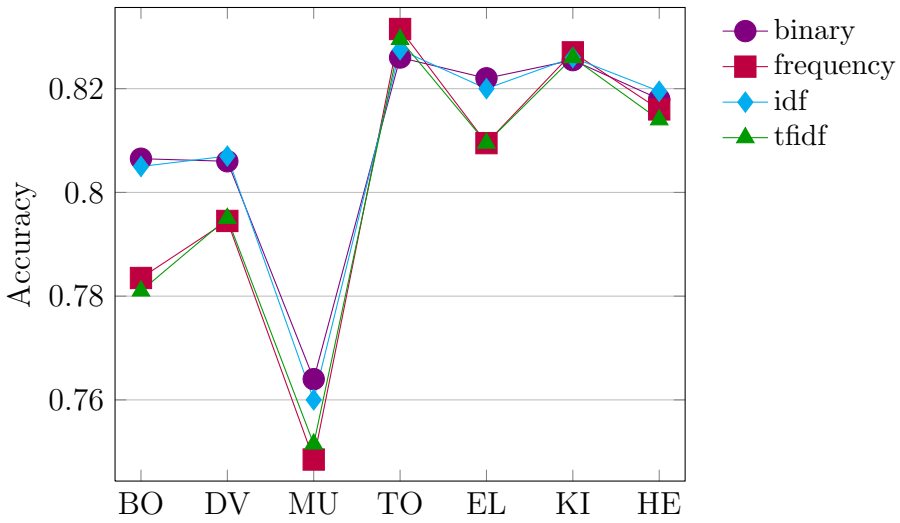

Figure 3.4: Feature weight impact on the in-domain accuracies (binary case)[10].

Third, experiments with tokens, lemmas and stems showed the advantage of tokens and stems over lemmas (Figure 3.5). Lemmas outperformed tokens and stems only for MU, but for the rest of the domains they only give modest results. This implies that word morphology is of substantial benefit to sentiment classification and should not be discarded. Stems and tokens provided comparable accuracies, although there is a slight superiority of stems over tokens, which is especially evident for HE. It is worth noting

[10]For presentation purposes and despite the fact that the values on the x-axis are categorical, data points are joined with lines. This style is used in all similar graphs.

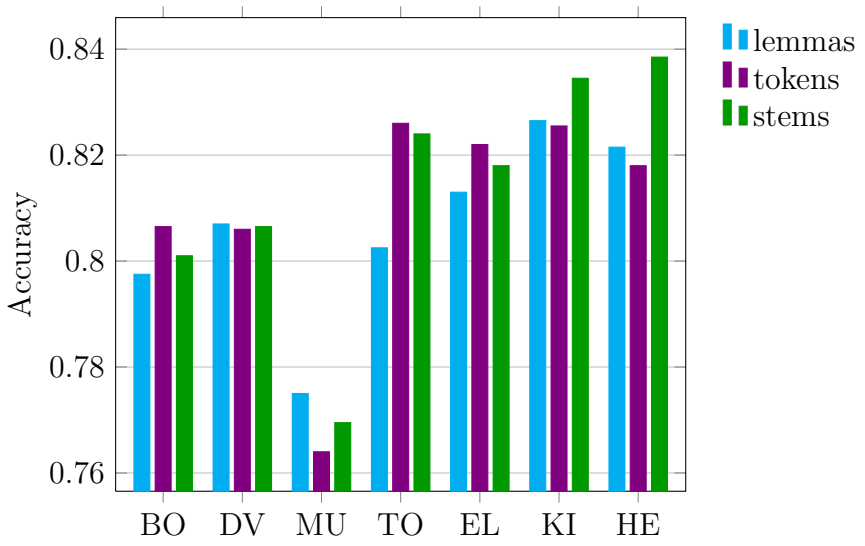

Figure 3.5: Word form impact on the in-domain accuracies (binary case).

that all experiments revealed that lexically richer domains, i.e. BO, DV and MU, yield a lower performance compared with TO, EL, KI and HE.

Finally, we applied the IG, WLLR and LR feature reduction techniques to different feature sets, varying weighting schemes, n-gram size, word forms and feature cut-off thresholds (from 250 to 5000). Some of our experimental results agreed with the full feature set performance: unigrams+bigrams outperform unigrams, frequency weights give similar results to tf-idf weights and the same is true for idf and binary weights. Due to this similarity, we do not show the accuracies obtained from unigrams weighted with frequency and idf. Figure 3.6 illustrates the accuracies after the IG feature reduction for different feature weights, word forms and feature numbers across all domains[11]. The x-axis corresponds to the number of features from 250 to

[11] Word forms weighted with different weighting schemes are described using the

5000. In contrast to the full feature set, binary tokens perform worst, and the best results are given by tf-idf lemmas and tf-idf stems. Therefore, when fewer features are involved, the information about their frequency becomes significant and morphological variations can be ignored. This is due to the decrease of feature redundancy which becomes important for a limited number of features. The behaviour of the accuracy graphs differs from one domain to another. In most cases, there is a maximum at around 500-1000 features and then the overall performance declines.

The optimal number of features is between 500-750 with a slight preference for 750 features. To establish the optimal feature reduction parameters valid for any data, we averaged the accuracies given by each feature set across all domains and selected the combination of features and the cut-off threshold providing the maximum average performance. As a result, we found two most effective feature sets: 750 tf-idf stems and 500 tf-idf lemmas.

following template "weighting scheme + word form". For example, tokens with binary weights are further called "binary tokens" while lemmas with with tf-idf weights are called "tf-idf lemmas".

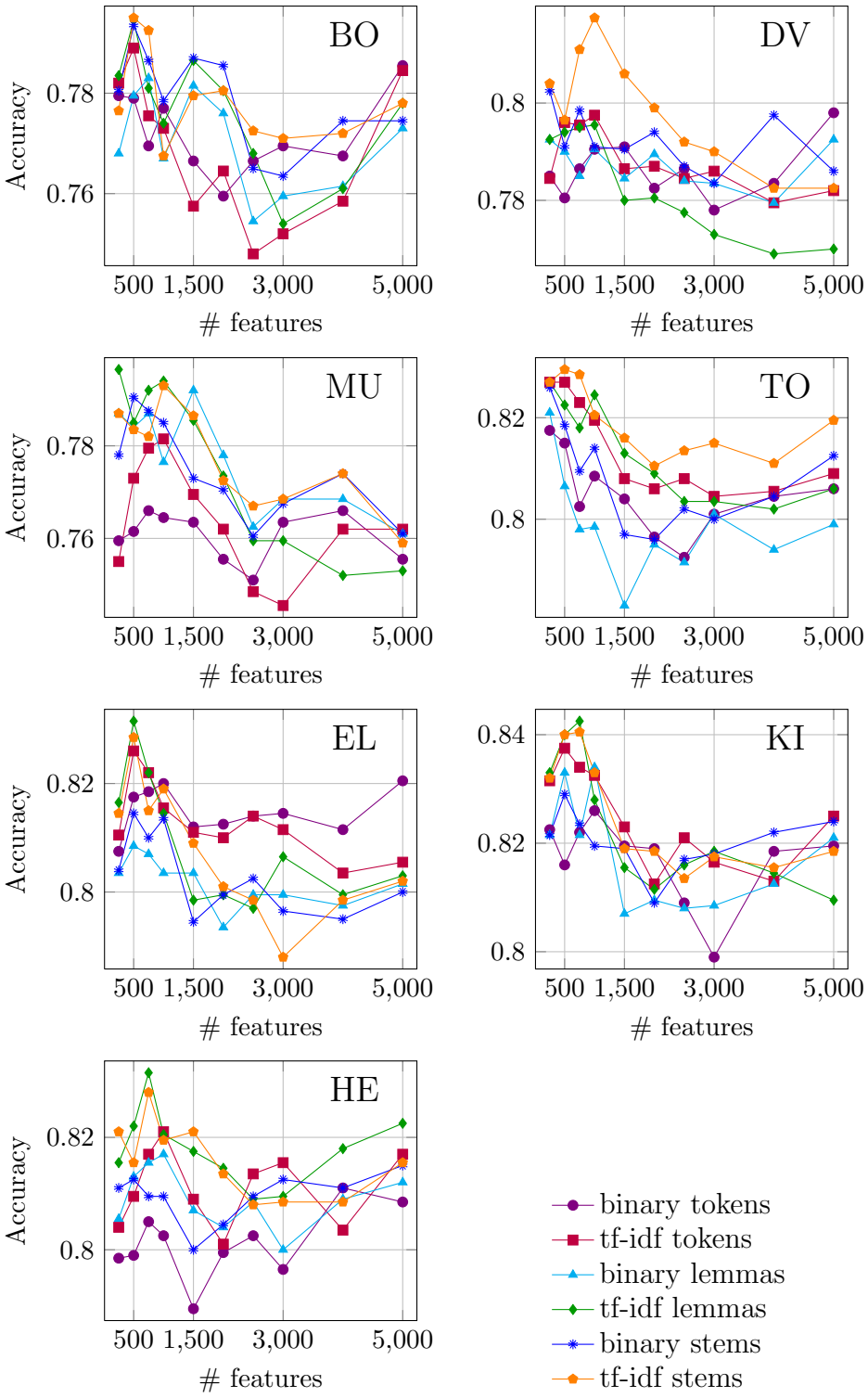

Figure 3.6: IG feature selection (binary case).

As far as other feature reduction techniques are concerned, WLLR exhibits very similar results, achieving the best accuracy in the interval of 500-750 features[12]. As expected, LR feature reduction reaches maximum accuracy for a large number of features (2500-3000 features), which makes it not very effective for feature reduction[13].

To compare accuracies delivered by different feature reduction techniques, we chose their two best combinations of feature weights, word forms and the number of features and depicted the corresponding accuracies for each domain in the same plot (Figure 3.7). LR feature reduction is the least successful since it delivers either the worst or mediocre results. However, IG and WLLR are equally effective, outperforming one another for certain domains. For example, WLLR in a combination with 750 tf-idf stems leads to a considerably better accuracy than IG with 750 tf-idf stems for BO and MU while it performs much worse for TO and KI. The paired t-test showed that the difference for the domain of KI is statistically significant. Therefore, there is a slight advantage of IG over WLLR. In summary, our exploration revealed that the most beneficial feature selection technique is largely determined by the data.

Finally, we investigate whether feature reduction improves the best performance reached when all features are considered. A comparison of the results obtained with the best feature reduction and full feature set is

[12] Due to the similarity of the results given by IG and WLLR feature reduction, the corresponding graphs for WLLR are omitted.

[13] The corresponding graphs for LR are not presented, due to the inability of LR to effectively filter features.

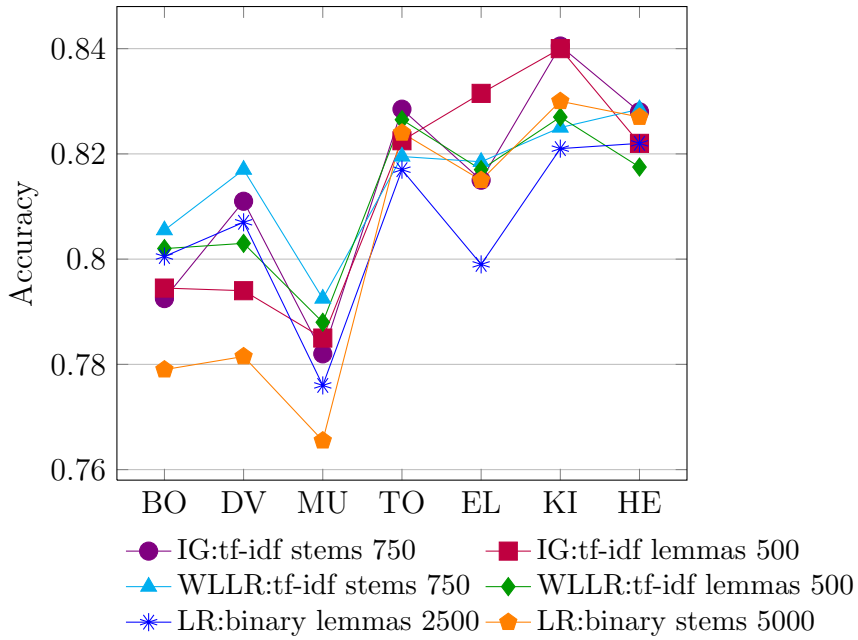

Figure 3.7: Comparison of feature selection techniques: IG, WLLR and LR (binary case).

given in Figure 3.8. We can observe that, overall, feature reduction does not substantially improve classification accuracy. The only domain where it appears to be very beneficial is MU, as the corresponding discrepancies in accuracies are statistically significant. In fact, MU is the only domain whose performance drastically degrades when the number of features increases (see Figure 3.6).

3.4.3 Multiclass classification

As the optimal features might change when more sentiment classes than in the binary case are involved, we carried out a new feature selection study for multiclass sentiment classification (Figures 3.9 and 3.10). In agreement

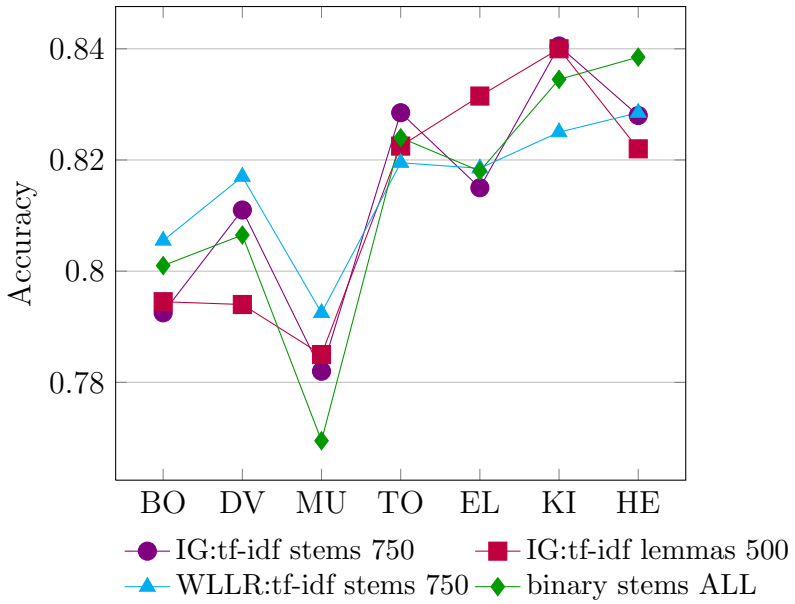

Figure 3.8: Comparison of the feature selection with the full feature set results (binary case).

with the binary case, unigrams+bigrams outperform unigrams, although the differences in accuracy are much smaller overall and disappear completely for EL and TO. Since frequency and tf-idf weights, as well as binary and idf weights, give almost identical results, we only report the accuracies achieved with binary and tf-idf weights. Figure 3.10 shows the accuracies computed for different combinations of feature weights and word forms. If the word form is fixed, then binary weights either outperform tf-idf weights or give comparable results. In agreement with the binary case, the difference between binary and tf-idf weights is highest for BO (Figure 3.4). Therefore, when the full feature set is exploited, information about feature frequency is unnecessary and, at times, detrimental as it degrades performance. We found no evidence that

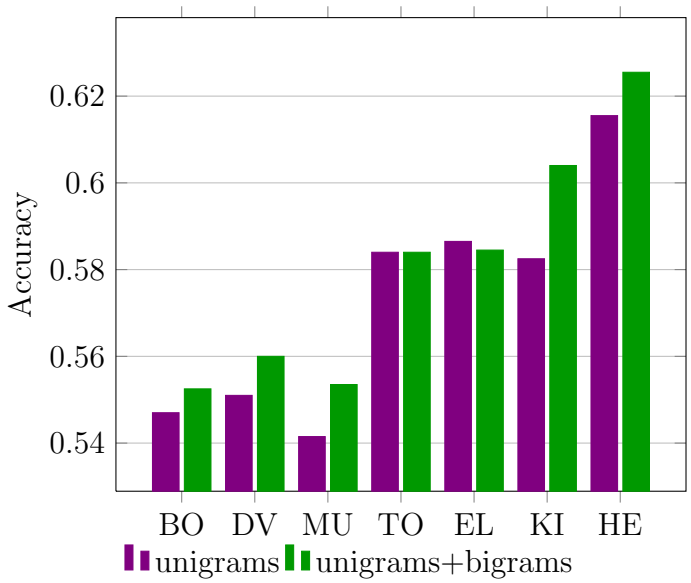

Figure 3.9: In-domain accuracies based on unigram and unigram+bigram vector representation (multiclass case).

accuracy depends on any particular word form. There is a slight tendency for binary stems and lemmas to surpass tokens, but these differences are not statistically significant.

We conducted feature reduction with the IG scoring function as it performed well in the binary case (Figure 3.11). The lexically richer domains of BO, DV and MU seem to benefit most from aggressive feature reduction as the accuracy levels for 250-500 features are comparable to the accuracy for the full dataset. The majority of feature reduction graphs have a tendency for a local maximum between 250 and 1000 features, which never surpasses the accuracy achieved with a higher number of features (except for the MU domain). This is different from the binary case, where aggressive

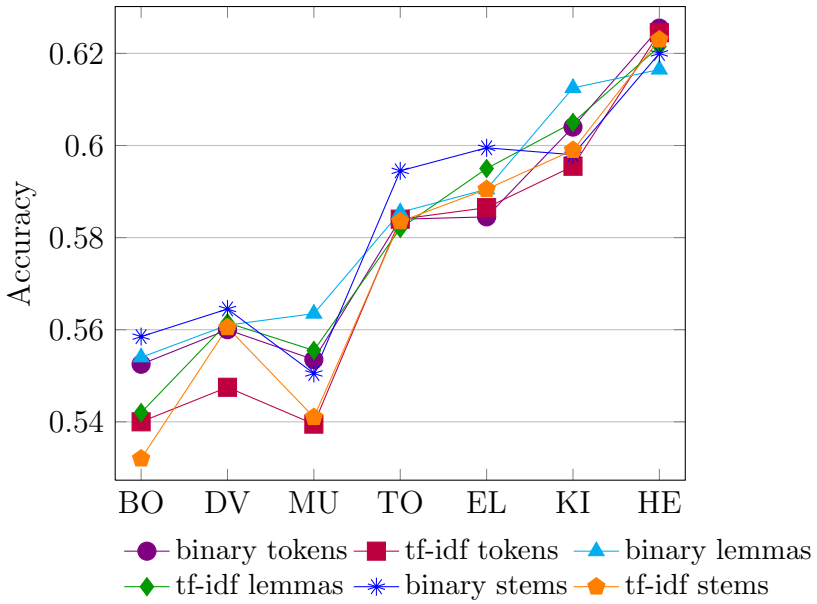

Figure 3.10: Feature weight and word form impact on the in-domain accuracies (multiclass case).

feature reduction (up to 500-750 features) is more effective and leads to a performance similar to the full feature set (Figure 3.8). We also observe that feature frequencies do not provide additional information on the sentiment strength of documents because tf-idf and binary weights give similar results. This is one possible reason why aggressive feature reduction is not successful for the multiclass case. In contrast to feature frequencies, word forms have a more pronounced impact on the accuracy. In particular, stems deliver the best results in most cases. This becomes especially clear when the number of features reaches 5000. Figure 3.12, where the highest accuracies after feature reduction and with the full feature set are displayed, confirms that multiclass classification does not benefit from feature reduction.

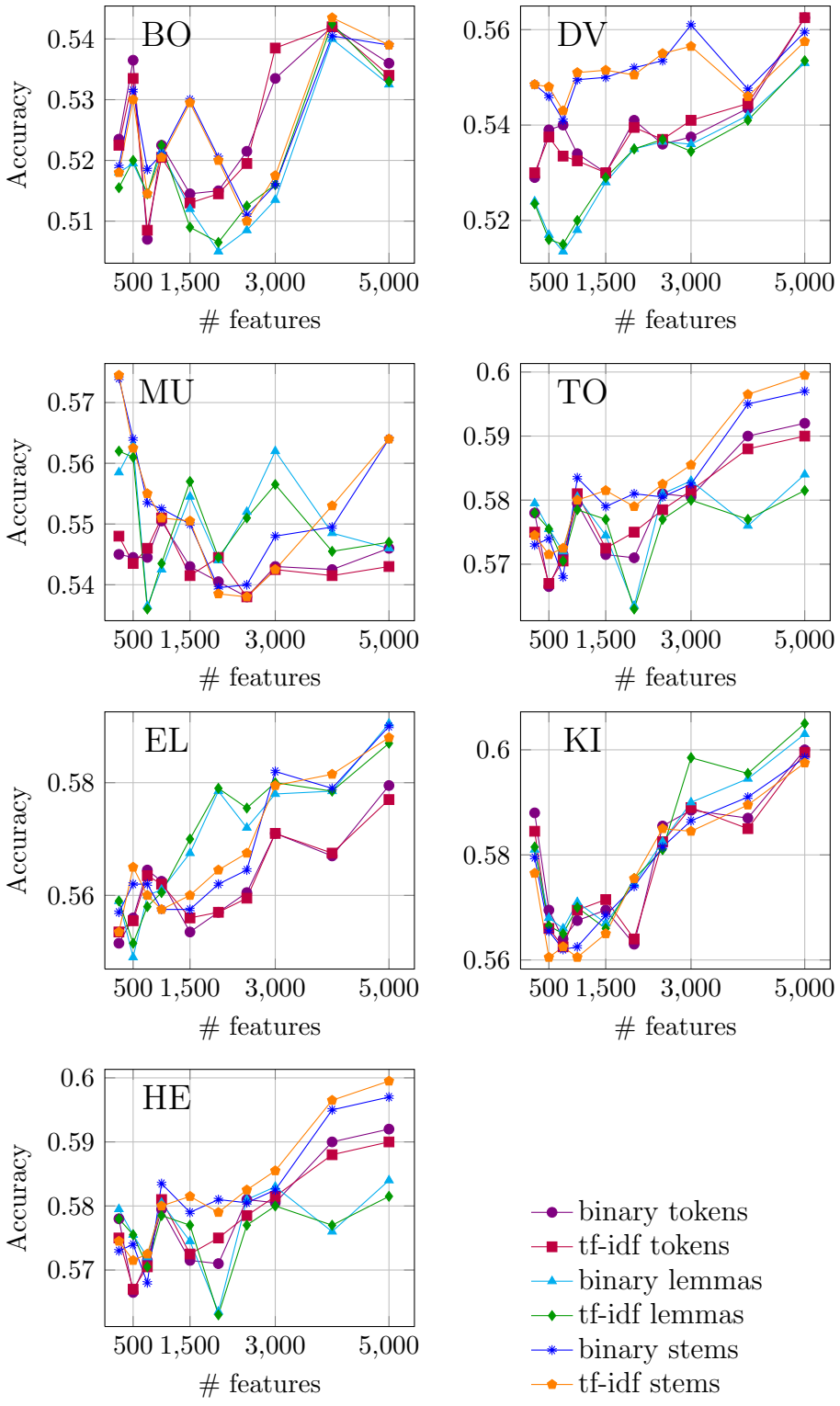

Figure 3.11: IG feature selection (multiclass case).

3.4.4 Discussion

The majority of our findings on feature selection are shared by both binary and multiclass cases. Our experiments revealed the following:

- Unigrams+bigrams always perform better than unigrams alone.
- Feature reduction does not improve the sentiment classification results.
- When the full feature set is exploited, feature frequencies in individual texts do not substantially improve performance compared with binary weighting and are sometimes detrimental.
- In most cases, stems marginally outperform the other word forms.

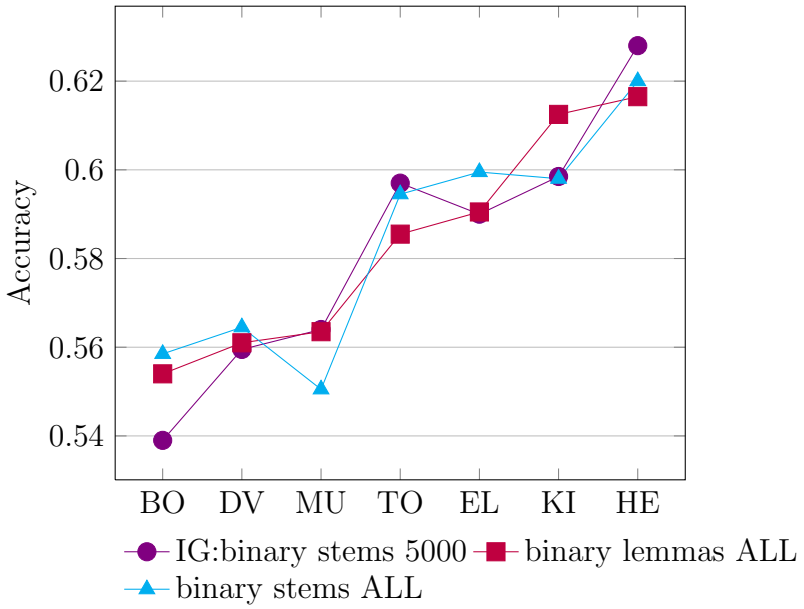

Figure 3.12: Comparison of the feature selection with the full feature set results (multiclass case).

Nevertheless, binary and multiclass sentiment classification exhibit certain differences:

- Tf-idf weights in combination with feature reduction demonstrate high performance for the binary case, but multiclass classification does not benefit from the information about feature frequencies.

- Aggressive feature reduction, which is shown to be relatively effective for binary classification, does not work well for multiclass classification with the exception of the MU domain.

- As expected, the binary accuracies are much higher than the multiclass accuracies due to the increased complexity of the multiclass task.

The best results (maximum accuracies from Figures 3.8 and 3.12) give us the accuracy upper bounds for all domains (Table 3.4)[14]. We also report the corresponding the $macroF_1$ values, which are much lower than the accuracies. Moreover, there is no correlation between $macroF_1$ and accuracy, that is, higher accuracy levels do not necessary imply higher $macroF_1$ levels. The upper bounds are used for comparison with the best semi-supervised and cross-domain accuracies and F-scores in Chapters 6 and 7.

[14] The multiclass results for the HE domain are much higher than those for any other domain. This phenomenon was studied and it was revealed that HE contains many duplicated reviews. However, we do not have an explanation why this did not influence the binary results, therefore, a more detailed study is required.

task		BO	DV	MU	TO	EL	KI	HE
binary		0.807	0.818	0.794	0.830	0.832	0.840	0.839
multi-class	accuracy	0.559	0.565	0.564	0.597	0.600	0.613	0.628
	$macroF_1$	0.435	0.453	0.438	0.460	0.453	0.440	0.498
	$\bar{F}_1$	0.497	0.509	0.501	0.529	0.527	0.527	0.563

Table 3.4: Performance upper bounds for the binary and multiclass tasks.

3.5 Baselines

In this section, we provide baselines for the binary and multiclass tasks in two experimental settings: semi-supervised and cross-domain. For multiclass classification, both baseline accuracies and $macroF_1$ are reported. All baselines are built using the full feature set because feature reduction is not appropriate either for the semi-supervised setting, due to the limited amount of labelled data and therefore a much smaller feature set, or for the cross-domain setting, due to the difference in feature sets for source and target domains.

3.5.1 Semi-supervised baselines

The baselines for semi-supervised classification were computed separately for each domain. We used a 5-fold cross-validation setup to conduct our evaluation. Thus, 400 examples were used for testing and the remaining 1600 instances were treated as a pool for a randomly selected labelled dataset. We gradually incremented the amount of labelled data from 50 to 800 examples to analyse their impact on the performance.

To compute the binary classification baseline we used binary stems

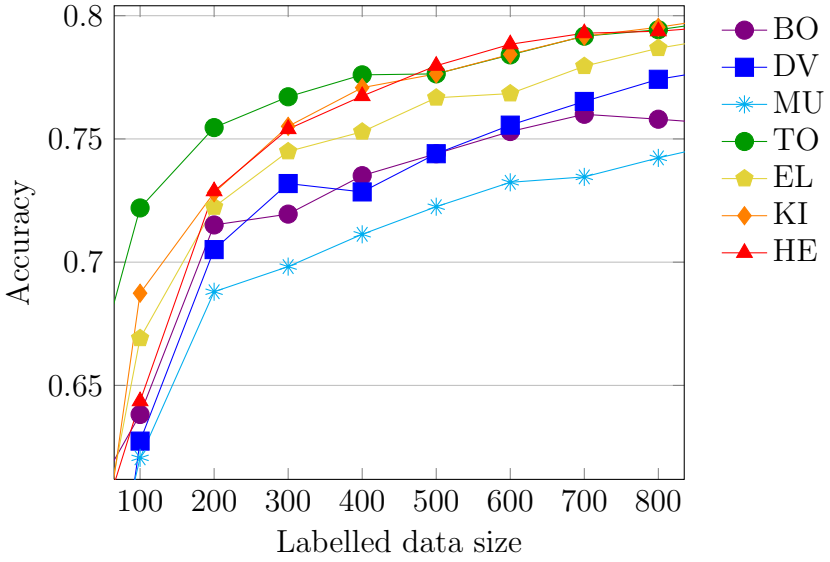

Figure 3.13: Semi-supervised baselines (binary case).

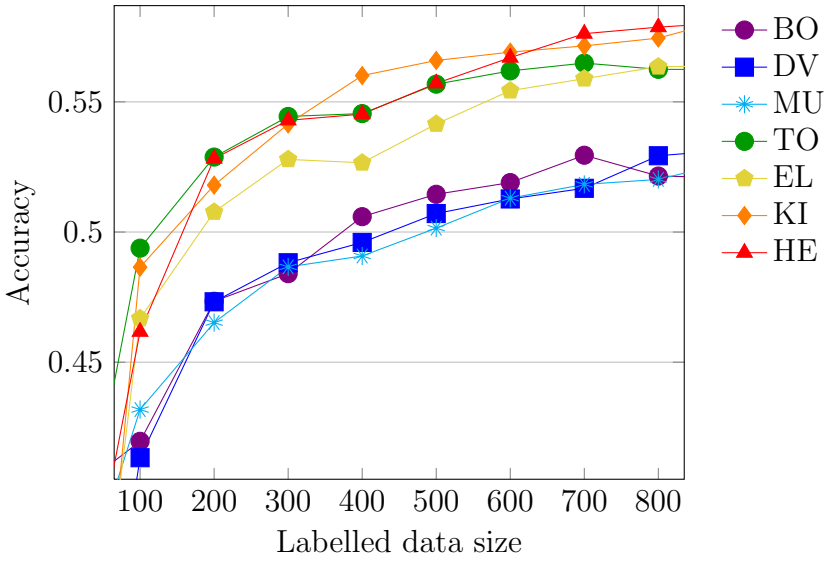

Figure 3.14: Semi-supervised baseline accuracies (multiclass case).

since this feature combination performed best in supervised settings (Figure 3.13). For multiclass classification, binary lemmas and binary stems yielded

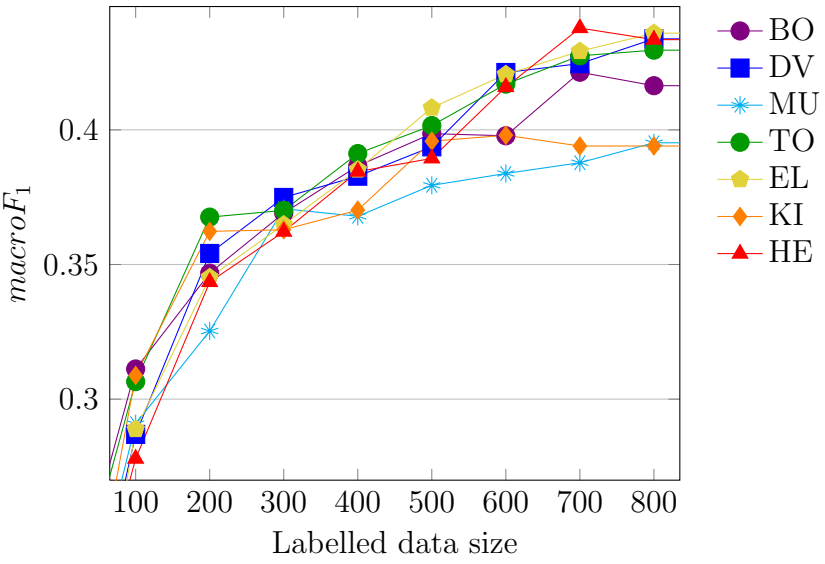

Figure 3.15: Semi-supervised baseline $macroF_1$ values (multiclass case).

the highest results (Figures 3.14 and 3.15) and we selected the latter feature combination to match the binary case. The difference in accuracies between lexically richer and lexically poorer domains is larger when more sentiment classes are involved. This means that higher task complexity makes the classification of more difficult data even harder. Due to the unbalanced class distribution, baseline $macroF_1$ values are overall 10-15 percentage points (ppt) lower than accuracies. Interestingly, there is not much difference between the $macroF_1$ graphs of lexically richer and poorer domains. Therefore, the higher accuracies for lexically poorer domains are due to the correct identification of strongly positive and negative sentiment classes, which have the larger sizes.

3.5.2 Cross-domain baselines

As with semi-supervised settings, we built our cross-domain baselines using binary stems as features. The seven datasets give 42 combinations of domain pairs. Figures 3.16-3.18 display the cross-domain baseline accuracies and $macroF_1$, where each graph corresponds to the same target dataset. We can identify two groups of target domains for which the cross-domain accuracies show different behaviour: BO, DV, MU and EL, KI, HE. The best results for the group of target domains BO, DV and MU are achieved when the source domain is also from this group. The same is true for the group of domains EL, KI and HE. In contrast, when the source data is from the opposite group of domains, the results are substantially lower.

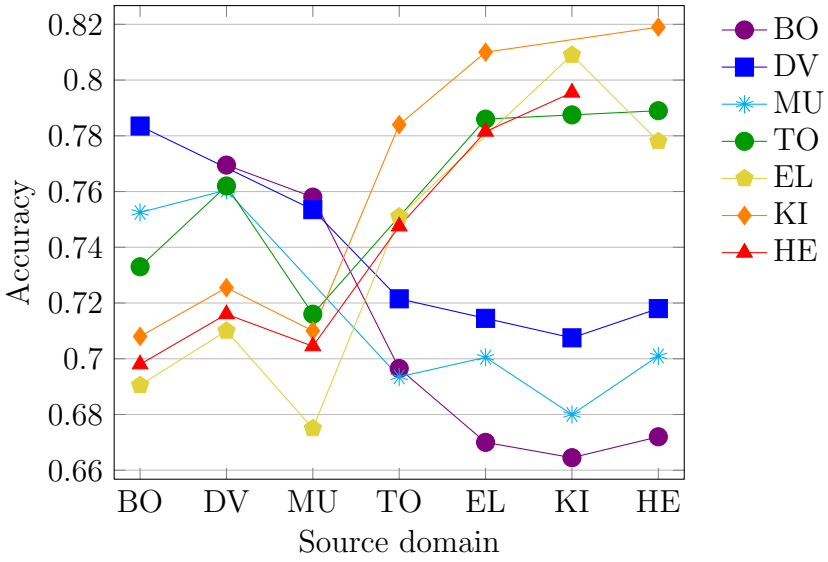

Figure 3.16: Cross-domain accuracy baselines where each curve corresponds to the same target dataset (binary case).

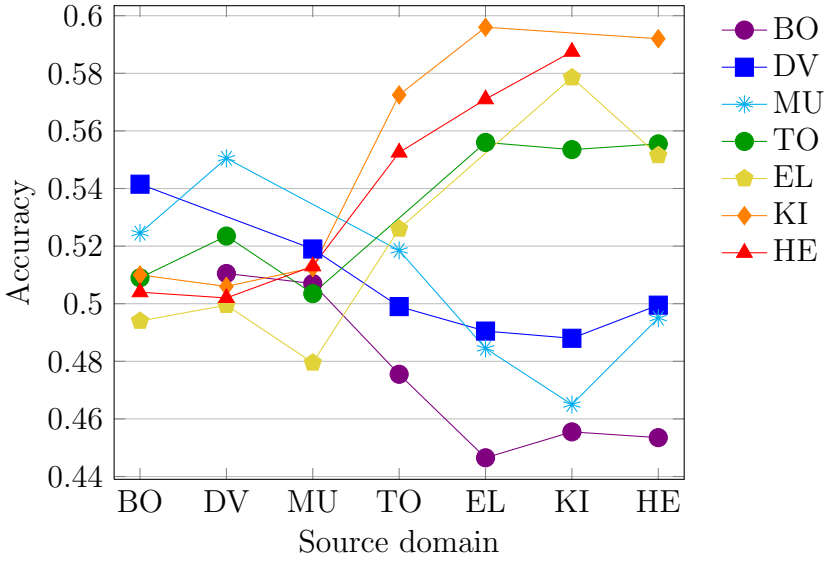

Figure 3.17: Cross-domain accuracy baselines where each curve corresponds to the same target dataset (multiclass case).

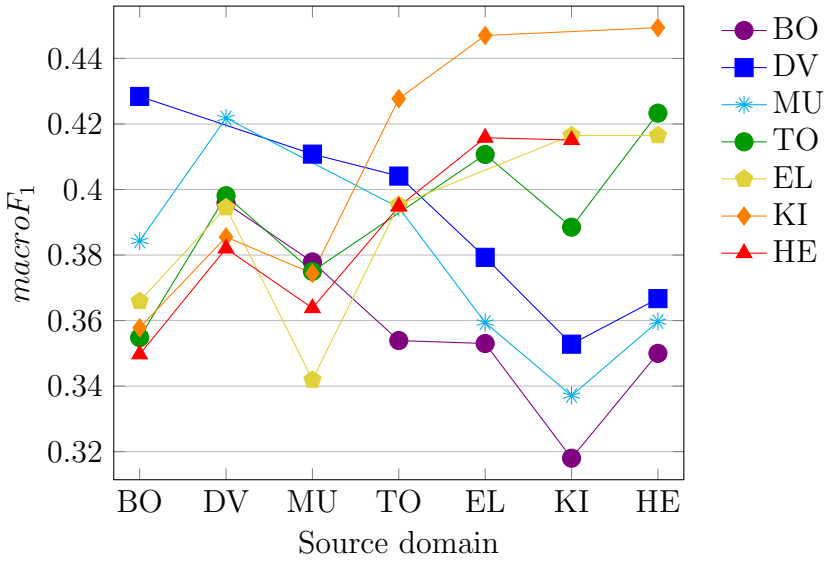

Figure 3.18: Cross-domain $macroF_1$ baselines where each curve corresponds to the same target dataset (multiclass case).

BO	DV	EL	KI
highly recommend	superb	plenty	so easy
concise	love it	highly recommend	worry
excellent book	really good	very happy	perfect for
for anyone	very well	please with	favorite
excellent	highly recommend	too_NOT	recommend it
unique	rare	worry	highly recommend
read for	well worth	recommend it	be excellent
my favorite	fascinating	be perfect	my only
must read	recommend it	well as	fiestaware
and also	perfectly	had_NOT	wonderful
life and	beautifully	happy with	soup
wonderful	great job	very easy	perfectly
resource	capture	works	excellent
power of	be excellent	awesome	stuff
insightful	really enjoy	glad	nicely
great book	mix	nicely	satisfy
be definitely	outstanding	price and	clean up
be easy	you love	sturdy	have_NOT to
will find	life and	i love	fun
overview	good movie	better than	serve

Table 3.5: 20 discriminative positive features ranked by likelihood.

To understand the reason behind the observed difference between the two groups of domains, we extracted the 20 most discriminative positive and negative features (unigrams + bigrams of tokens) from the BO, DV, EL and KI datasets using the LR scoring function (Tables 3.5 and 3.6). Since the LR metric tends to overvalue rare features, we removed features occurring less than 20 times in each dataset. A feature comparison across domains reveals that some domains share more similarities than others. For example, features such as “refund”, “return”, “repair”, “defective” and “customer service” appearing in both EL and KI are highly unlikely for BO and DV. At the

BO	DV	EL	KI
disappointing	your money	poorly	waste of
poorly	so bad	refund	return it
enough_NOT	worth_NOT	repair	recommend_NOT
waste of	ridiculous	waste of	it break
your money	waste of	waste_NOT	refund
annoying	worst movie	never buy	worth_NOT
page to	waste_NOT	defective	to return
buy_NOT	pointless	forum	very disappoint
bunch	recommend_NOT	stop work	worst
boring	lame	junk	your money
just do	horrible	worst	do work_NOT
to finish	may have	do buy_NOT	do buy_NOT
unless	boring	horrible	exchange
hope that	why do	return	it back
let me	awful	customer service	poorly
waste_NOT	suck	your money	repair
i keep	dull	mistake	send it
to believe	i hate	be tell	useless
nothing to	do even_NOT	very disappoint	warranty
recommend_NOT	garbage	send it	defective

Table 3.6: 20 discriminative negative features ranked by likelihood.

same time, the features “boring”, “dull”, “fascinating”, and “enjoy” that are common in BO and DV, are quite improbable for KI and EL. Concerning the rest of the datasets, MU reviews are close to BO and DV, and HE reviews are close to EL and KI. The TO domain is in between these two groups, since it shares similarities with both of them. For instance, games can be boring like books or movies but they might also be defective and sent back like electronic devices or kitchen appliances. However, Figures 3.16 and 3.17 indicate that TO reviews share more similarities with EL, KI and HE.

3.6 Summary

In this chapter, we first described the multi-domain dataset of product reviews used in our experiments. Then, in Section 3.2, we gave an overview of the main preprocessing steps, which, among others, contain a rule-based treatment of negation - an important procedure for sentiment classification. In Section 3.3, we listed the evaluation metrics to be used including accuracy, MSE, $macroF_1$ and $\bar{F}_1$ (the mean between the accuracy and $macroF_1$). The last two metrics are needed because of the imbalanced class distribution of the multiclass data and will assess whether the representation of sentiment classes is balanced in the final results. The feature selection study reported in Section 3.4 showed the advantage of unigrams+bigrams over unigrams alone and a marginal advantage of stems over other word forms. The result of feature reduction was found to be negative because it did not improve performance compared to the full feature set. Finally, in Section 3.5, the baselines for semi-supervised and cross-domain sentiment classification were computed with linear kernel SVM as the baseline classifier.

Chapter 4

Graph-based learning

The chapter describes a graph-based learning approach adopted as the main method of this thesis and its application to sentiment classification. Graph-based learning exploits the ability of the data to be represented as a weighted graph where instances are vertices and edges reflect similarities between instances. It assumes that strongly connected instances tend to belong to the same class (the manifold or smoothness assumption). For sentiment classification, instances are documents and document similarity is used to indicate the closeness of document sentiments. Graph-based learning is discriminative, non-parametric and transductive. The last property can be seen as a shortcoming as transductive models are designed for closed datasets only and cannot handle unseen examples. However, various studies suggest different ways of transforming transductive learners into inductive ones (Zhu et al., 2003b; Chapelle et al., 2002; Delalleau et al., 2005; Sindhwani et al., 2005)[1].

There are several arguments to support our choice for graph-based learning. First, it can be easily applied to both semi-supervised and cross-domain tasks without any change in the implementation. This puts semi-

[1]We intend to address this issue as part of our future work.

supervised and cross-domain techniques under the same conditions and makes comparisons fairer. Second, the extension of graph-based algorithms to multiclass classification is straightforward, which is crucial for many real-world scenarios. Third, graph-based algorithms perform well for many NLP tasks and are a competitive alternative to other semi-supervised and cross-domain techniques. At the same time, however, there is a lack of a thorough comparison between graph-based algorithms and other cross-domain and semi-supervised methods regarding their application to sentiment classification. This thesis aims to fill that gap and also to investigate how different algorithm parameters and modifications to the graph structure impact on classification accuracy. Finally, graph-based algorithms can be easily scaled to solve large problems with millions of instances. In particular, Bilmes and Subramanya (2011) proposed a graph node reordering heuristic and demonstrated its effectiveness on a huge graph with 120 million nodes. In this thesis, we do not study scalability issues as our data is of a much smaller size. However, this property is very important for real-world problems where the ability to deal with as much data as possible in an efficient way can help to perform both accurate and fast classification.

4.1 Notation and problem setting

First, we introduce the following notation for the formal problem setting:

- $X = \{\mathbf{x}_i\}_{i=1}^{n}$ is a dataset of n documents where $\mathbf{x}_i$ refers to the document vector in the vector space model[2].
- l documents are labelled and u are unlabelled, $l + u = n$.
- Without a loss of generality we consider a binary classification problem.
- $\{y_i\}_{i=1}^{l}$ is a set of probabilistic labels corresponding to labelled documents.
- y_i indicates the probability of a document $\mathbf{x}_i$ belonging to class 1. Similarly, $(1 - y_i)$ is the probability that $\mathbf{x}_i$ belongs to class 0[3].
- $\{\hat{y}_i\}_{i=l+1}^{n}$ are unknown labels that must be induced.

In graph-based learning, labelled and unlabelled instances are jointly represented as an undirected weighted graph $G = (V, E, W)$. Vertices $V = \{1, ..., n\}$ correspond to n data points X, connected through edges $E \subseteq V \times V$, where $L = \{1, ..., l\}$ are labelled and $U = \{l + 1, ..., n\}$ are unlabelled vertices, $V = L \bigcup U$. $W = (w_{ij})$ is a weight matrix indicating the similarity between a pair of nodes connected by the respective edge. Graph-based learning requires that the entries of W are non-negative and symmetric, which is true for the similarity measure introduced in Section 4.2.2. The task is to assign probabilities $\hat{y}_i$ to unlabelled nodes U.

[2] In Section 4.2.2, different document representations will be discussed.

[3] If a multiclass classification problem with m classes is considered, a document label represents an m-dimensional vector $y_i = (y_{i1}, ...y_{im})$ where each entry y_{ik} indicates the probability of $\mathbf{x}_i$ belonging to class C_k.

When applying graph-based learning to sentiment classification, two important questions must be addressed:

1. *How do we construct a sentiment graph?* This is key for the successful performance of graph-based learning (Zhu, 2005). It poses a further question regarding the similarity measure between documents that expresses the closeness of document sentiments rather than content. Our solution to this problem is given in Section 4.2.

2. *Which inference algorithm should be used?* Many graph-based algorithms have been proposed recently: graph mincuts (Blum and Chawla, 2001), label propagation (LP) (Zhu and Ghahramani, 2002) and its derivatives (Zhu et al., 2003a; Zhou et al., 2004; Goldberg and Zhu, 2006), spectral graph transducer (Joachims, 2003), manifold regularisation (Belkin et al., 2006), modified adsorption (Talukdar and Crammer, 2009) and measure propagation (Subramanya and Bilmes, 2011). In this study, we use LP together with its modifications due to its simplicity, robustness, high performance and extensive exploitation in other NLP tasks.

The remainder of the chapter is organised as follows: Section 4.2 explains our method of constructing sentiment graphs. In Section 4.3, the LP algorithm is described and in Section 4.4, two ways of balancing skewed output labels are reviewed. The LP modifications used in our experiments

are presented in Section 4.5. Finally, Section 4.6 gives an overview of the main stages of our graph-based sentiment analysis system.

4.2 Sentiment graph construction

Many studies on graph-based learning emphasise the significance of graph construction. Zhu (2008, page 18) argued that "it is more important to construct a good graph than to choose among the methods". Bilmes and Subramanya (2011, page 14) pointed out that "the graph determines how information flows from one sample to another and thus an incorrect choice of a neighborhood can lead to poor results". In this section we consider two aspects of graph construction: graph connectivity and similarity metrics.

4.2.1 Graph connectivity

Graphs can be fully connected or sparse. Fully connected graphs, besides their high computational cost, usually perform worse than sparse models (Zhu, 2005). The most common way to construct sparse graphs is to introduce either a parameter for the number of nearest neighbours, k, to each vertex (kNN graphs) or a maximum proximity radius, ϵ, that delimits connected nodes of a vertex from other graph nodes (ϵNN graphs). According to Zhu (2005), all kNN graphs tend to perform well empirically. Following this observation as well as our own experiments with ϵNN graphs, which showed no significant difference in performance, we choose kNN graphs to represent our data.

Unlike the vast majority of studies where the number of neighbours k is considered the same for all vertices (Zhu, 2005; Talukdar and Crammer, 2009), we follow the work of Goldberg and Zhu (2006) and distinguish between labelled and unlabelled nodes, connecting each unlabelled node with k_l labelled and k_u unlabelled neighbours, $k_l \neq k_u$. Our experimental results (Chapters 6 and 7) demonstrate the importance of this modification.

Note that the weight matrix of kNN graphs is not necessarily symmetric due to the non-symmetry of the nearest neighbourhood relation. A good example of this is a hub node: it is a nearest neighbour to many nodes but only k of them can be its nearest neighbours. To make the matrix W symmetric, we require the following condition: for any two nodes i and j, if the node i is a nearest neighbour of the node j, then the node j must be a nearest neighbour of the node i. Let $k_uNN(i)$ be a set of k_u nearest unlabelled neighbours and $k_lNN(i)$ be a set of k_l nearest labelled neighbours of the node i. Then, the weight matrix $W = (w_{ij})$ must satisfy (4.1).

$$w_{ij} = \begin{cases} w_{ij} & \text{if } j \in k_uNN(i) \cup k_lNN(i) \\ w_{ij} & \text{if } i \in k_uNN(j) \cup k_lNN(j) \\ 0 & \text{otherwise} \end{cases} \tag{4.1}$$

4.2.2 Sentiment similarity

Sentiment classification requires a similarity metric which assigns values to a pair of documents on the basis of their sentiment, so that documents with

the same sentiment have high similarity scores and documents with opposite sentiment have low scores. This implies that the vector representation of documents must contain sentiment markers rather than topic words. Previous research suggests several ways to tailor document representation to the purposes of sentiment similarity. Pang and Lee (2005) proposed PSP-based similarity, representing documents as (PSP, 1-PSP), where PSP is the percentage of positive sentences in a document. They used an additional classifier for learning sentence polarity that was trained on external data with user-provided scores. As a result, the PSP values had a high correlation with document ratings. Goldberg and Zhu (2006) also used in-domain labelled data to approximate sentiment similarity for semi-supervised multiclass classification. They constructed a vector representation based on document words where the weight of words was computed using their mutual information with positive and negative classes from an external labelled dataset. The main disadvantage of both approaches is that they rely on labelled in-domain data, which can be very expensive to obtain. In contrast, the principal purpose of our research is to create a framework where only a limited amount of labelled data is available. Other work on graph-based learning for sentiment analysis did not focus on the problem of sentiment similarity and used a straightforward approach by considering documents via vectors of their words (Wu et al., 2009; Talukdar and Crammer, 2009).

Following Goldberg and Zhu (2006) and Pang and Lee (2005), we examine two types of document representation: *document feature-based* and *document*

unit-based, although unlike the above studies no external labelled data is required. Moreover, for each type of document representation, several ways of computing it are suggested. Those are then further verified and compared to select the representation which best reflects document sentiments. Document similarity is then estimated as the cosine similarity between corresponding document representations.

4.2.2.1 Document feature-based representation

In this representation, we express a document as a vector of features that can potentially convey sentiment. The question here concerns the features to be selected. According to previous studies, adjectives, verbs and adverbs are good indicators of sentiment (Pang and Lee, 2008). At the same time, nouns can also express positive or negative feelings, for example, "problem", "laughter", "mercy", although they are most likely to be topic words. To distinguish subjective and topic nouns we extend our feature set with nouns listed in the SO-CAL dictionaries, which are manually created lexicons of sentiment words rated from -5 (very negative) to 5 (very positive) and grouped by parts of speech (Taboada et al., 2011). Since the context of words is an important clue to their sentiment, we also enhance the feature set with bigrams containing relevant parts of speech.

4.2.2.2 Document unit-based representation

The unit-based representation describes a document through the sentiment values of its units. Several types of document units are considered:

words, sentences, groups of sentences (for example, three first and three last sentences of a document), document titles, and, finally the document itself. Finer-grained units (words and sentences) are used to estimate the sentiment strength of coarse-grained units (the whole document, groups of its sentences and the document title), which are assumed to reflect the same sentiment as the document itself. We assess the effectiveness of the following representations:

- PSP, NSP - percentage of positive/negative sentences in a document[4].
- PWP, NWP - percentage of positive/negative words in a document.
- SentWP (Sent1WP, Sent2WP,...) - percentage of words corresponding to the sentiment strengths *Sent1, Sent2,...* according to a sentiment lexicon scale.
- 3FirstPWP, 3FirstNWP - percentage of positive/negative words in the first three sentences.
- 3LastPWP, 3LastNWP - percentage of positive/negative words in the last three sentences.
- TitlePWP, TitleNWP - percentage of positive/negative words in the title.

Positive and negative words are extracted using the SO-CAL dictionaries. For more accurate computing of unit-based representations, we employ the

[4]Note that NSP $\neq$ 1- PSP as we allow some sentences to be neutral.

negation module described in Section 3.2. We also experimented with other sentiment resources such as SentiWordNet (Esuli and Sebastiani, 2006b) and SentiStrength (Thelwall et al., 2012), but the former gives unsatisfactory performance, while the latter, focusing on explicit sentiment, is insufficient for review data and finds no sentiment in a substantial number of documents.

An error analysis revealed that many errors in the scores of unit-based representations are caused by specific words that do not bear sentiment in general contexts but signal a strong attitude towards a product in consumer reviews. Such words are especially frequent in negative contexts, for example, "return", "refund", "avoid" and "break". We also identified verbs that imply negative sentiment towards a product only when used with negations, for example, "buy_NOT", "work_NOT". Evidently, these cannot be considered as expressing sentiment in a strict sense, but they do imply a clear user feeling about a product and, therefore, serve as sentiment markers. Since these cases are quite frequent in reviews we decided to adapt the SO-CAL dictionaries to fit the product review data.

4.2.2.3 Adapting SO-CAL to review data

Our adaptation algorithm aims to select words which typically bear sentiment in the context of product reviews and, thus, are discriminative features for sentiment classification. This means that we can use one of the feature selection metrics presented in Chapter 3. We found the likelihood ratio (LR) to be the most relevant for the task as it has the highest discriminative

power in comparison with information gain and weighted log-likelihood ratio. The likelihood ratio overestimates rare and specific features but this can be overcome by using substantial amounts of data.

The new sentiment markers are searched for in a large dataset of product reviews that include all seven domains of interest (BO, DV, MU, TO, EL, KI and HE). Each domain is represented by 20k documents, half of which are positive and half negative. It is worth noting that this data has been compiled independently from the test and training dataset described in Section 3.1. For each domain, words outside the SO-CAL dictionaries with LR $\geq$ 1.5 are extracted, which means that they are 1.5 times more frequent in the texts of one polarity than in the texts of the opposite polarity. This gives seven lists of domain-specific sentiment markers. Finally, to acquire only domain-independent sentiment markers, the seven lists are compared and words appearing in at least three domains and having an average LR $\geq$ 2 are selected[5] and manually rated by an expert on a scale from -5 to 5. As a result, we obtained 65 new sentiment markers (46 negative and 19 positive) including "refund", "return", "buy_NOT", "work_NOT", "send" and "even_NOT", identified as mostly negative, and "highly", "family", "fast", "storage" and "overall", identified as mostly positive. The full list of sentiment markers together with their strengths and LR scores, averaged over domains, can be

[5] We also added several words with $1.9 \leq LR \leq 2$ which were found in more than five domains.

found in Appendix A. Table 4.1 shows the number of new sentiment markers that were found to be both frequent and discriminative for each domain.

We also present distributions of the positive and negative sentiment markers separately. Although we discovered many more negative than positive sentiment markers, their ratio is almost equal for the domains of BO, DV and MU. Overall, less than half of the new sentiment markers are frequent in the lexically richer domains. This proportion is much higher for the lexically poorer domains and varies from 68% for EL to 78% for TO. This suggests that the SO-CAL adaptation will have a greater impact on the lexically poorer domains.

Sentiment markers	# of sentiment markers per domain						
	BO	DV	MU	TO	EL	KI	HE
all	31	32	31	51	44	49	45
positive	12	15	16	11	7	11	10
negative	19	17	15	40	37	38	35

Table 4.1: Distribution of the new sentiment markers over domains

The sentiment dictionary expansion revealed some interesting facts about sentiment markers. Only six sentiment markers were found to be common for all seven domains: positive "highly" and negative "unless", "maybe", "buy_NOT", "money" and "even_not". Surprisingly, the word "money" and its different representations such as "dollar" and "$", are mainly negative, while the word "price" is mostly positive. In general, negative markers are more numerous and more discriminative than positive markers. Some negative markers imply a clear opinion about a product

and, therefore, were rated as strong sentiments, for example, "return", "refund" and "work_NOT". The sentiment strength of positive markers is not as pronounced, for example, "include", "provide" and "bring" and, therefore, their sentiment scores are moderate. Interestingly, mentions of relatives are most likely to occur in positive reviews: "family" (LR=2.4), "husband" (LR=2.1), "wife" (LR=1.8), "brother" (LR=1.8). Finally, we found cases of polarity switching for several sentiment markers, for example, the word "today" appears as positive in BO (LR=1.7), DV (LR=2.0) and MU (LR=2.3) reviews and as negative in EL (LR=-2.5), KI (LR=-1.8) and HE (LR=-2.1) reviews. The negative connotation of "today" for EL, KI, and HE can be explained by numerous examples when devices break and users immediately share their negative experience to warn others: "Today, March 6th, I am returning it and this is why" or "Today is April 6 and the thing is broken". Sometimes, "today" is used in positive contexts, for instance,"I like it so much, I'm buying a second one today" but, due to the specificity of the EL, KI and HE domains these cases are not very frequent. In the reviews for books and movies (which rarely break and need to be returned) people tend to use "today" when they have sentimental feelings about a product and/or compare it with those of today: "This was one of my favs as a kid and I still love it today" or "They don't make films like this today". Occasionally the comparison is not in favour of a reviewed product, for example, "The movie, by today's standards, is a little disjointed and incomplete", but these cases were quite rare in our dataset.

4.2.2.4 Evaluation of similarity metrics

Feature-based and unit-based vectors are independent and can be complementary. The former can be viewed as a domain-specific representation because it consists of sentiment-bearing words used in a given domain. In turn, the latter can be regarded as a domain-independent or generic representation, since the SO-CAL dictionaries are domain-independent and, thus, scores of document units do not preserve any domain knowledge. To benefit from both domain-specific and generic information, we integrate feature-based and unit-based representations. Due to the different dimensions of feature-based and unit-based vectors, we combine similarities corresponding to each representation rather than the vectors themselves. The similarity metric obtained is called *hybrid*.

Denote by F_i and U_i the feature-based and unit-based representations of a document i, respectively. Then, the hybrid similarity sim_H between documents i and j is the following:

$$sim_H = sim(F_i, F_j) + sim(U_i, U_j), \tag{4.2}$$

where $sim(\cdot, \cdot)$ stands for the cosine similarity between its arguments.

Our evaluation of the similarity metrics exploits the principal premise of graph-based learning: smoothness of the label function on the graph. This implies that documents in the nearest neighbourhood have similar sentiment labels. We propose an evaluation metric called Δ_y, which computes the

average difference between document sentiments y_i in a neighborhood:

$$\Delta_y = \frac{1}{k \cdot n} \sum_{i \in V} \sum_{j \in kNN(i)} |y_i - y_j| \tag{4.3}$$

where $kNN(i)$ are the k nearest neighbours of a document i and $n = |V|$ is the number of graph nodes. It is presumed that the quality of a similarity metric is determined by the value of Δ_y and the best metric conforms to the minimum Δ_y value.

Table 4.2 presents our evaluation results with $k = 100$[6] for the different similarity metrics. To assess which combination of document representation components performs best, they are ordered by their impact on the minimisation of Δ_y from highest (PWP) to lowest (3FirstPWP) and are added one by one into the document representation. The impact of each component is determined on the basis of its individual performance. These different combinations are then compared with two baselines: the feature-based representation alone and all components of the unit-based representation.

The results of Table 4.2 can be summarised as follows:

- The feature-based representation alone gives the poorest results.
- The most effective unit-based components, which have the highest impact on the quality of the similarity measure, are PWP, TitlePWP and PSP.
- Not all unit-based components improve the quality of the similarity measure when considered in combination with other components.

[6]Other values of k were also tested and they all gave similar results.

Document representations	Δ_y - average document neighborhood sentiment difference						
	BO	DV	MU	EL	KI	TO	HE
PWP	1.031	0.932	0.968	1.016	1.065	1.007	1.136
+TitlePWP	0.928	0.893	0.923	0.883	0.861	0.852	0.895
+PSP	0.909	0.887	0.901	0.872	0.841	0.834	0.856
+Feature-based	**0.898**	**0.870**	**0.893**	**0.846**	**0.819**	**0.807**	**0.835**
+3LastPWP	0.910	0.879	0.902	0.862	0.827	0.818	0.850
+SentWP	0.913	0.880	0.900	0.865	0.834	0.823	0.859
+3FirstPWP	0.919	0.874	0.903	0.883	0.846	0.833	0.874
Feature-based	1.191	1.173	1.184	1.144	1.149	1.119	1.175
All Unit-based	0.926	0.884	0.904	0.899	0.864	0.847	0.888

Table 4.2: Evaluation of the similarity metrics based on different document representations ("+" adds the corresponding component to all those above it; the best combination is highlighted).

While the quality of the similarity measure consistently improves when PWP, TitlePWP, PSP and the feature-based representation are included, it decreases moderately with every subsequent component (3LastPWP, SentWP and 3FirstPWP) for all domains.

- Although the feature-based representation does not perform well alone, adding it to the best combination of unit-based components provides the lowest Δ_y.

We can conclude that according to our intrinsic evaluation, the hybrid similarity metric based on document features and PWP, PSP and TitlePWP components is the most accurate estimate for sentiment similarity. Therefore, we use this measure in the evaluation of our graph-based sentiment analysis system described in Chapters 6 and 7.

4.3 Label Propagation

LP was one of the first graph-based algorithms to be developed (Zhu and Ghahramani, 2002). It is an iterative process that at each step propagates information from labelled to unlabelled nodes until convergence, i.e., when node labels do not change from one iteration to another. *LP* is the weighted averaging of labels in a node neighbourhood where the influence of neighbours is set by edge weights (Figure 4.1). Therefore, the smoothness of the data is presumed by the algorithm.

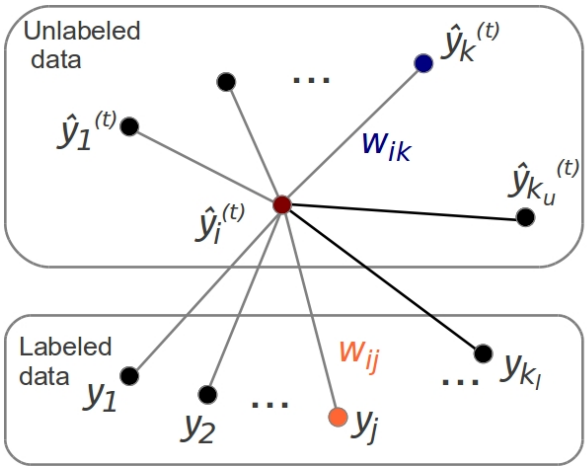

Figure 4.1: Graph structure for *LP*

We denote by $\bar{W} = (\bar{w}_{ij})$ a row-normalised version of the matrix W, referred to as the transition matrix:

$$\bar{w}_{ij} = \frac{w_{ij}}{\sum_j w_{ij}} \tag{4.4}$$

LP is described in Algorithm 1[7].

Algorithm 1. LP

- Compute weight matrix W (see Section 4.2)

- Compute transition matrix $\bar{W}$ using (4.4)

- Initialise $\hat{Y}^{(0)} = (y_1, ..., y_l, 0, ..., 0)$, $t = 0$

- Iterate

  1. $\hat{Y}^{(t+1)} \leftarrow \bar{W}\hat{Y}^{(t)}$

  2. $\hat{Y}_l^{(t+1)} \leftarrow Y_l$[8]

until convergence to $\hat{Y}^{\infty}$

The transition matrix $\bar{W}$ can be split into four sub-matrices:

$$\bar{W} = \begin{pmatrix} \bar{W}_{ll} & \bar{W}_{lu} \\ \bar{W}_{ul} & \bar{W}_{uu} \end{pmatrix} \tag{4.5}$$

Zhu and Ghahramani (2002) proved the convergence of Algorithm 1 to the following simple solution:

$$\hat{Y_u} = (I - \bar{W}_{uu})^{-1}\bar{W}_{ul}Y_l \tag{4.6}$$

LP can be formalised using the optimisation framework. Indeed, graph-based learning may be seen as finding a labeling function consistent with

[7]For multiclass classification with m classes the algorithm must be run m times in a one-vs-all fashion.

[8]Y_l is reset on each iteration in order to clamp it to the initial value.

the graph structure and smoothness assumption. This can be expressed by minimisation of the quadratic cost function which penalises strongly related nodes with different labels:

$$C(\hat{Y}) = \sum_{ij} w_{ij}(\hat{y}_i - \hat{y}_j)^2 \rightarrow min \tag{4.7}$$

The optimisation problem (4.7) and Algorithm 1 are equivalent and have identical solutions given by (4.6) (Zhu, 2005; Bengio et al., 2006).

We use two rules (referred to as *probability combination rules*) to induce a sentiment class of a document i from its output class probabilities $\hat{y}_i = (\hat{y}_{i1}, ..\hat{y}_{ik}, ..\hat{y}_{im})$. The first rule, called the maximum probability rule ($maxP$), is a straightforward approach which assigns the class corresponding to the maximum value of $\hat{y}_{ik}$. The second rule, called hierarchical ($HIER$), applies the maximum probability rule in a hierarchical manner and can only be used for the multiclass case. First, it finds a node polarity by comparing sums of probabilities for positive and negative sentiment classes. All sentiment classes whose polarity is opposite to the one established are discarded from further consideration. Then, the maximum probability rule is used on the remaining sentiment classes to induce the output node sentiment. The hierarchical probability combination rule could help to address errors related to incorrect sentiment polarity, which are more serious than errors in sentiment strength.

4.4 Balancing class proportions

Zhu et al. (2003a) pointed out that if classes are not well-separated then the

final distribution of labels can be highly skewed. However, if class priors are known, the output labels can be modified to match the class proportions. Various methods have been developed to learn the class priors of test data, for example, the expectation maximisation algorithm (Saerens et al., 2001) and Pearson divergence minimisation (du Plessis and Sugiyama, 2012). In this thesis it is assumed that the training and test datasets have similar class distributions (which is especially true in semi-supervised settings) and, therefore, that class priors can be estimated from the labelled data.

Following Zhu and Ghahramani (2002) two post-processing techniques for class balancing are adopted: class mass normalisation (CMN) and label bidding (LB).

- **Class mass normalisation (CMN)** scales output probabilities so that the final class masses match the priors. Prior probabilities p_0 and p_1 of classes 0 and 1 can be estimated from the labelled data:

$$p_0 = \frac{1}{l}\sum_{\forall i \in L}(1 - y_i), \quad p_1 = \frac{1}{l}\sum_{\forall i \in L} y_i \tag{4.8}$$

  Similarly, we compute the output class masses m_0 and m_1 of classes 0 and 1 over the output probabilities of the unlabelled data:

$$m_0 = \frac{1}{u}\sum_{\forall i \in U}(1 - \hat{y}_i), \quad m_1 = \frac{1}{u}\sum_{\forall i \in U} \hat{y}_i \tag{4.9}$$

  Using formulas (4.8) and (4.9), we can define the following decision rule for $\hat{y}_i$ to belong to class 1:

$$\frac{p_1\hat{y}_i}{m_1} > \frac{p_0(1-\hat{y}_i)}{m_0} \tag{4.10}$$

The direction of inequality (4.10) is reversed for the decision rule regarding class 0.

- **Label bidding** (LB) maintains the invariability of prior class sizes rather than prior class probabilities. Suppose that our labelled data consists of l_0 examples from class 0 and l_1 examples from class 1. Then, the output of LB normalisation will contain $\frac{l_0 u}{l}$ examples of class 0 and $\frac{l_1 u}{l}$ examples of class 1. LB can be seen as u-step process. During each step, an element with the highest probability, $\hat{p}_i = \max\{1-\hat{y}_i, \hat{y}_i\}$, is selected and added to the corresponding class. When one of the classes reaches its maximum number of elements, all remaining examples are assigned to the opposite class.

4.5 LP modifications

In this section we describe several modifications of LP. One of them, LP_γ, introduces a parameter γ, which gives a different weight to labelled and unlabelled neighbours. Another, $LP_{\alpha\beta}$, incorporates predictions from external classifiers into the graph structure.

4.5.1 LP_γ: Weighting labelled and unlabelled neighbours

The graph structure used in LP (Figure 4.1) does not differentiate between labelled and unlabelled neighbours. However, it might be beneficial to give them different weights. For example, in semi-supervised graph-based learning it is natural to rely more on labelled neighbours whose labels are identified with a high level of confidence. In contrast, for cross-domain learning, highly reliable labelled nodes might not help much if the source and target data are very different. In such cases, the contribution of unlabelled examples should be increased.

We use a parameter $\gamma \in (0, 1)$ to control the distribution of weights between labelled and unlabelled examples, so that $\gamma < 0.5$ gives preference to unlabelled and $\gamma > 0.5$ to labelled neighbours (Figure 4.2A). We formalise this LP variant in Algorithm 2.

Wu et al. (2009) proposed a similar algorithm, called graph ranking, and applied it to cross-domain sentiment classification. This method has two main differences to LP_γ. First, the weight matrices W_{uu} and W_{ul} are normalised separately instead of using the same scaling factor for labelled and unlabelled data. This difference has no effect as long as the scaling factors for both matrices are similar. However, this might not be the case for cross-domain graphs. Indeed, if source and target domains are very different so that out-of-domain neighbours are much farther away than in-domain neighbours, the scaling factors can have different orders of magnitude.

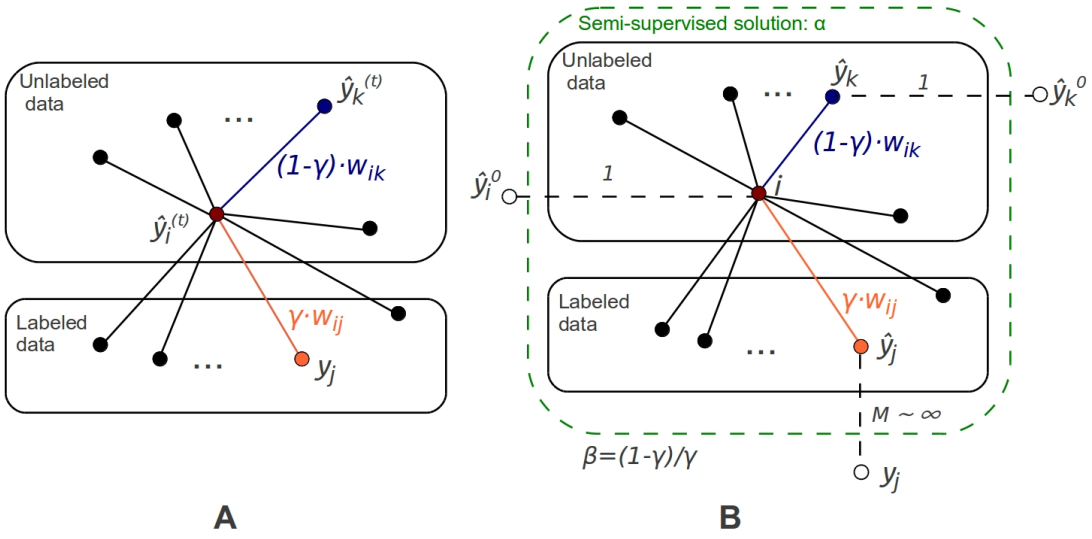

Figure 4.2: LP modifications: **A** Different weight for labelled and unlabelled neighbours (LP_γ); **B** Incorporating external predictions ($LP_{\alpha\beta}$).

Algorithm 2. LP_γ

- Compute weight matrix W

- Compute transition matrix $\bar{W}$ using (4.4)

- Build sub-matrices $\bar{W}_{uu}$ and $\bar{W}_{ul}$ as in (4.5)

- Initialise $Y_l = (y_1, ..., y_l)$, $\hat{Y}_u^{(0)} = (0, ..., 0)$

- Choose a parameter $\gamma \in [0, 1]$

- Iterate

$$\hat{Y}_u^{(t+1)} \leftarrow (1-\gamma)\bar{W}_{uu}\hat{Y}_u^{(t)} + \gamma\bar{W}_{ul}Y_l$$

until convergence to $\hat{Y}_u^{\infty}$

Second, the updated values of unlabelled nodes are normalised using CMN after each iteration to match the class priors. The method of Wu et al. (2009) (referred to as $RANK$ to acknowledge its original name) is presented in Algorithm 3.

Algorithm 3. $RANK$

- Compute weight matrix W
- Build sub-matrices W_{uu} and W_{ul} similar to (4.5)
- Compute normalised matrices $\bar{W}_{uu}$ and $\bar{W}_{ul}$ by

  $\bar{w}_{ij}^{uu} = \frac{w_{ij}^{uu}}{\sum_j w_{ij}^{uu}}$ and $\bar{w}_{ij}^{ul} = \frac{w_{ij}^{ul}}{\sum_j w_{ij}^{ul}}$

- Initialise $Y_l = (y_1, ..., y_l)$, $\hat{Y}_u^{(0)} = (0, ..., 0)$
- Choose a parameter $\gamma \in [0, 1]$
- Iterate

  1. $\hat{Y}_u^{(t+1)} \leftarrow (1-\gamma)\bar{W}_{uu}\hat{Y}_u^{(t)} + \gamma\bar{W}_{ul}Y_l$
  2. Normalise $\hat{Y}_u^{(t+1)}$ with CMN

until convergence to $\hat{Y}_u^{\infty}$

4.5.2 $LP_{\alpha\beta}$: Incorporating external classifiers

We can further modify the graph structure by incorporating predictions given by external classifiers. Zhu et al. (2003a) introduced additional nodes

(called "dongle" nodes) for storing initial predictions and connected them to unlabelled vertices. A similar idea was implemented by Goldberg and Zhu (2006), who applied graph-based learning to semi-supervised multiclass sentiment classification. In their modification, both labelled and unlabelled vertices are connected to dongle nodes, which allows noise in labelled data (Figure 4.2B). The degree of confidence of the values of labelled data y_i is controlled by the parameter M (Figure 4.2B). A higher value of M corresponds to a higher confidence of y_i and a lower chance of $\hat{y}_i$ being different to y_i. Similar to LP_γ, this modification exploits the parameter γ to control the weights of labelled and unlabelled neighbours. The algorithm seeks the solution that minimises the discrepancies between:

1. output values of unlabelled nodes $\hat{y}_i$, $i \in U$ and output values of their labelled and unlabelled neighbours $\hat{y}_j$, $j \in L \cup U$ (smoothness condition);

2. output values of unlabelled nodes $\hat{y}_i$ and their supervised predictions y_i^0, $i \in U$;

3. output values of labelled nodes $\hat{y}_i$ and their initial labels y_i, $i \in L$.

Taking into consideration the above conditions, the problem can be formulated as a minimisation of the following cost function:

$$\mathcal{C}(\hat{Y}) = \sum_{i \in L} M(\hat{y}_i - y_i)^2 + \sum_{i \in U} (\hat{y}_i - y_i^0)^2 + \sum_{i \in U} \sum_{j \in k_l NN(i)} \gamma w_{ij} (\hat{y}_i - \hat{y}_j)^2 + \sum_{i \in U} \sum_{j \in k_u NN(i)} (1-\gamma) w_{ij} (\hat{y}_i - \hat{y}_j)^2 \rightarrow min \tag{4.11}$$

After the substitutions $\alpha = \gamma k_l + (1-\gamma) k_u$ and $\beta = \frac{1-\gamma}{\gamma}$, the final optimisation problem can be written as:

$$\mathcal{C}(\hat{Y}) = \sum_{i \in L} M(\hat{y}_i - y_i)^2 + \sum_{i \in U} \left[(\hat{y}_i - \hat{y}_i^0)^2 + \frac{\alpha}{k_l + \beta k_u} \left(\sum_{j \in k_l NN(i)} w_{ij} (\hat{y}_i - \hat{y}_j)^2 + \sum_{j \in k_u NN(i)} \beta w_{ij} (\hat{y}_i - \hat{y}_j)^2 \right) \right] \rightarrow min \tag{4.12}$$

This LP variant (called $LP_{\alpha\beta}$) is able to take advantage of different sources of information. It relies on two main parameters, α and β. β is an analogue of γ in LP_γ: β close to 0 prioritises labelled neighbours (which corresponds to γ close to 1), while high values of β ($\beta \rightarrow \infty$) increase the weight of unlabelled neighbours (which corresponds to γ close to 0). Parameter α controls the weight of the graph-based solution compared to the initial predictions. α close to 0 gives more importance to the initial predictions, whilst high values of α prioritise the graph-based solution. For further details about the implementation of $LP_{\alpha\beta}$ see Goldberg and Zhu (2006).

4.6 The design of the classification module

The classification module is designed in accordance with the graph-based theory presented in this chapter. It includes three stages: graph construction, graph-based inference and post-processing (Figure 4.3). At the **graph construction** stage, the sentiment graph is created based on connectivity and the similarity measure given. Consequently, it requires the initialisation of two parameters responsible for graph connectivity, k_l and k_u, and the definition of the similarity measure, which can be achieved by choosing a relevant set of document representation components (see Section 4.2.2). When the sentiment graph is constructed, it is passed to the **graph-based inference** stage to perform sentiment classification. This stage requires a user to choose an LP variant out of four algorithms, presented in this chapter, LP, LP_γ, $LP_{\alpha\beta}$ and $RANK$, and to specify its parameters, α and/or β. As output, vectors of probabilities for all test documents are produced. Each entry of these vectors refers to the probability of a given document to be assigned to the corresponding class. Finally, the output values can be modified at the **post-processing** stage using two groups of techniques: normalisation (CMN or LB) and application of the probability combination rules ($maxP$ or $HIER$).

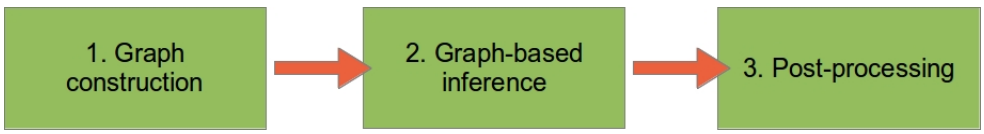

Figure 4.3: The main stages of the classification module.

4.7 Summary

This chapter described the graph-based learning approach adopted in the thesis. In Section 4.1, we introduced the notation used throughout the chapter and gave the formal definition of the graph-based classification problem. Section 4.2 focused on graph construction, which included establishing graph connectivity and estimating the sentiment similarity between documents. Based on previous research, we adopted kNN graphs instead of fully connected graphs as they proved to work best in practice. To estimate sentiment similarity, feature-based and unit-based document representations were proposed. In addition, the SO-CAL dictionaries used for computing document representations were adapted to the genre of product reviews. The intrinsic evaluation of the different similarity measures revealed the advantage of the hybrid measure, which uses both feature-based and unit-based representations. In Section 4.3, the LP algorithm was described and the two ways of combining its output probability values ($maxP$ and $HIER$) were proposed. Section 4.4 reviewed two normalisation techniques for fixing unbalanced distributions of output probabilities. Section 4.5 introduced three LP modifications, LP_{γ}, $LP_{\alpha\beta}$ and $RANK$, which attempt to improve the performance of LP. Finally, the design of our graph-based classification module was presented in Section 4.6.

Chapter 5

Study of data characteristics

In this chapter, we identify and analyse data characteristics which could influence the performance of machine learning methods in semi-supervised and cross-domain settings. The study of such characteristics may help to predict a learning output, which, in turn, could ease the problem of choosing the most appropriate data and learning setup. In the previous chapter, three phenomena were observed.

- **In-domain performance differs considerably from one dataset to another.** For example, our data gives up to a 5% difference in performance between the most accurate and the least accurate domains (Table 3.4). This implies that some datasets are more complex for learning than others and, thus, will require more labelled data to achieve the same performance than less complex datasets. In this chapter, we introduce the notion of domain complexity, which characterises the difficulty machine learning techniques have in learning sentiment classes on a given domain[1].

[1]It should be noted that our definition of domain complexity and its estimation serve the purposes of text classification and are not meant to be generalised for other machine learning problems.

- **Cross-domain results are highly dependent on source-target domain pairs.** This result is expected as the performance of machine learning techniques depends on how similar the training and test data are. This suggests that domain similarity is an important data characteristic and its estimation from the data could help in choosing the most appropriate training data for given test data and, perhaps, to predict the success of domain adaptation. In this chapter, we apply and test various well-known metrics from corpus linguistics and information theory which measure the distance (or similarity) between two datasets.

- **Sentiment performance drops drastically when extending the sentiment scale with two more sentiment classes.** Figures 3.13-3.17 show a decrease in accuracy of 20-25%. The problem of distinguishing between four classes is naturally harder than binary classification, therefore, some performance loss is expected. Yet it seems a high price to pay for only two additional classes. Another reason for the accuracy drop may be cases of a poor match between review ratings and texts when the ratings are not just binary (Carrillo de Albornoz et al., 2011; Maks and Vossen, 2013). Different users assign different meanings to 1*, 2*, 3*, etc. and, therefore, review ratings in a corpus of reviews written by different authors can be inconsistent. To study this issue in more detail, we design and conduct a human annotation experiment, consisting of the manual annotation of a

small number of product reviews by three native English speakers. The results of the annotators, reviewers and machine learning algorithms are then jointly analysed to serve the following purposes:

- to assess the complexity of the sentiment classification task with two and four sentiment classes;
- to analyse the conformity of review ratings;
- to estimate an upper bound of performance that is feasible for automatic methods to achieve.

The remainder of the chapter is organised as follows. In Section 5.1 we introduce the notion of domain complexity and suggest measures for its estimation. Based on the existing research, several measures of domain similarity are examined and those giving the highest correlation with the cross-domain accuracies are selected in Section 5.2. Finally, Section 5.3 describes the human annotation experiment and analyses the complexity of the sentiment classification task and the quality of review data. In all experiments of this chapter, our dataset with seven domains, BO, DV, MU, TO, EL, KI and HE, is exploited.

5.1 Domain complexity

We define domain complexity for text classification as a data characteristic that indicates how difficult it is to classify the data. Moreover, we are interested in data complexity rather than task complexity, therefore, data

labels are not taken into consideration. This means that such factors as the number of classes and distribution of classes in the data are not considered. Of course, multiple classes or skewed class distribution make learning from data harder but these properties are attributed to the task rather than to the data. If a classification task is fixed, all of these factors will have a similar influence on the classification performance and, thus, can be ignored.

Domain complexity can be determined using information from different levels: lexical, syntactic, semantic, discourse and pragmatic. As our sentiment classifier uses solely lexical features, we rely on lexical information only to define complexity. In Chapter 3, we observed that domains with the highest vocabulary richness also have the lowest in-domain accuracies. Moreover, vocabulary richness implies long tail probability distributions, which make it more difficult for statistical methods to learn from the data as they rely on frequently occurring features. These factors imply that domain complexity can be estimated through vocabulary richness. One of the established measures of vocabulary richness is TTR (Biber et al., 2002). In addition, we propose another measure, called the percentage of rare words, as it also indicates long tail probability distributions. We consider that a word is rare if it occurs in fewer than 10 documents[2]. The values for TTR and the percentage of rare words for all seven domains were previously given in Table 3.1 together with other data statistics.

In order to find out which of the vocabulary richness measures estimates

[2]This threshold was chosen following Tesitelova (1992).

domain complexity best, the Pearson product-moment correlation coefficient was calculated between each measure and the in-domain accuracies. As the results should be independent of the learning algorithm, we exploit two classifiers, linear SVM and voted perceptron (VP)[3]. To obtain a more reliable correlation coefficients[4], we artificially enhanced the number of datasets from 7 to 28, using different data sizes: 25%, 50%, 75% and 100% of the data. The correlation coefficients are presented in Table 5.1[5]. Although both domain complexity measures show high correlation with the in-domain accuracies, TTR delivers significantly higher correlation on average than the percentage of rare words.

Measure	SVM	VP	average
% of rare words	-0.829	-0.842	-0.835
TTR	-0.858	-0.927	-0.893

Table 5.1: Pearson correlation between the domain complexity measures and in-domain accuracies given by SVM and VP. The data comprises the seven domains of different sizes: 25%, 50%, 75% and 100% of the whole amount of data in a domain, resulting in 28 data points altogether.

Figure 5.1 depicts the relationship between the two complexity measures and the in-domain accuracies delivered by the SVM and VP classifiers. There is a clear boundary on both graphs, which separates simple and complex domains. For TTR it lies between 0.065 and 0.07 and for the percentage of

[3]We used the Weka (http://www.cs.waikato.ac.nz/ml/weka/) implementation of VP with default settings.

[4]Initially, seven domains give only seven data points for computing correlation.

[5] The negative correlation is due to the inverse relationship between accuracy and domain complexity: higher domain complexity implies lower classification accuracy.

rare words it is around 89%. Although these boundaries are valid for our data, it should not be taken for granted that they will be valid for all other datasets. In future, we plan to further justify the boundaries between simple and complex domains in a cross-validation setup exploiting a larger number of domains.

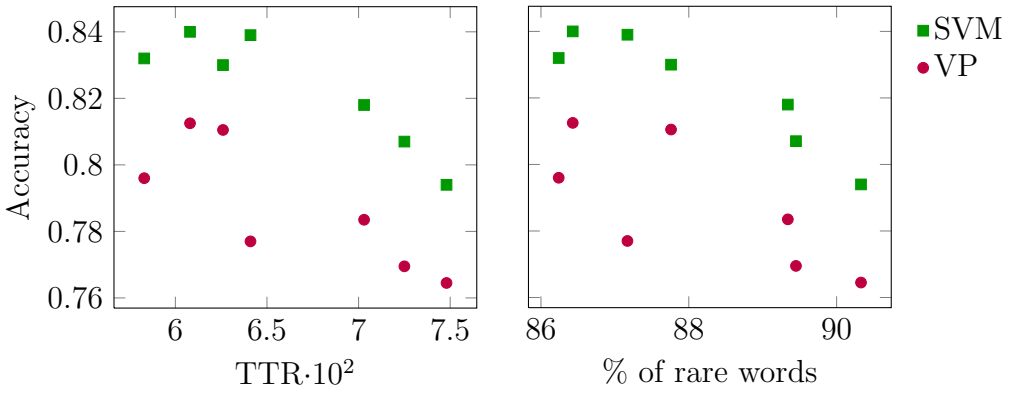

Figure 5.1: The relationship between complexity measures and in-domain accuracies given by SVM and VP.

5.2 Domain similarity

Domain similarity is an important characteristic of a domain pair when addressing the domain adaptation problem. Figures 3.16 and 3.17 show that cross-domain accuracies depend substantially on a domain pair and that they are higher for intuitively more similar domains and much lower for more distinct domains (Tables 3.5 and 3.6). Therefore, if several source datasets are available, measuring domain similarity could help to select the most appropriate data for training.

Domain similarity requires the definition of two components: features and

a similarity function. Domains are represented by corpora of documents, therefore, a feature representation of domains is equivalent to a feature representation of document collections. Since our goal is to estimate similarity in a sentiment sense, we select only features which are most likely to have a sentiment connotation. As for document similarity (Section 4.2.2), unigrams and bigrams containing adjectives, verbs and adverbs are considered. The chosen features are weighted with *idf* because this worked best for document similarity.

Possible similarity functions[6] mostly fall into two groups: probabilistically-motivated and geometrically-motivated functions (Plank and van Noord, 2011). Probabilistically-motivated functions consider a corpus as a distribution of its features and, therefore, the distance between two corpora is measured as the divergence between their feature distributions. Most of these functions are borrowed from statistics or information theory, for example, χ^2, Kullback-Leibler divergence (D_{KL}) (Kullback and Leibler, 1951) and its symmetric analogue Jensen-Shannon divergence (D_{JS}) (Lin, 1991), Renyi divergence (D_α) (Renyi, 1961) and many others. On the other hand, representing a corpus as a vector of its features, allows a plethora of different geometrically-motivated functions, such as cosine similarity (*cosine*), euclidean distance (L_2) and variational distance (L_1).

We select the following functions as candidates for estimation of domain

[6]We consider both similarity and distance measures as they can be obtained from each other by inversion.

similarity: *cosine*, L_1, L_2, χ^2, D_{KL}, D_{JS} and D_α. The choice of these functions was governed by previous studies in the field. Kilgarriff (2001) tested three functions for comparing corpora: χ^2, Spearman rank correlation coefficient and cross-entropy. χ^2 showed the best correlation with the gold standard. Asch and Daelemans (2010) examined cosine, L_1, L_2, D_{KL}, D_{JS} and D_α when estimating the performance loss of a PoS tagger across domains. The authors concluded that D_α with $\alpha = 0.99$ achieved the highest correlation with the cross-domain accuracies. Plank and van Noord (2011) came to a different conclusions when addressing the domain adaptation problem for parsing. For their task, D_{JS} and L_1 were found to be best, while D_α performed worst. However, in contrast to Asch and Daelemans (2010) and our research, Plank and van Noord (2011) used similarity measures to create labelled data similar to the target domain; therefore, they measured the similarity between a corpus and a document rather than between two corpora. This could explain why asymmetric measures, like D_{KL} and D_α were not very successful.

Let $S = \{s_i\}$ and $T = \{t_i\}$ be feature sets of source and target domains respectively. Then *cosine*, χ^2, L_1, L_2, D_{KL}, D_{JS} and D_α can be represented as follows:

$$cosine(S,T) = \frac{\sum_i s_i t_i}{\sqrt{\sum_i s_i^2 \sum_i t_i^2}} \tag{5.1}$$

$$\chi^2(S,T) = \frac{\sum_i (s_i - \bar{s}_i)^2}{\bar{s}_i} + \frac{\sum_i (t_i - \bar{t}_i)^2}{\bar{t}_i},$$
$$where: \tag{5.2}$$
$$\bar{s}_i = \frac{|S|(s_i + t_i)}{|S| + |T|}, \bar{t}_i = \frac{|T|(s_i + t_i)}{|S| + |T|}.$$

$$L_1(S,T) = \sum_i |s_i - t_i| \tag{5.3}$$

$$L_2(S,T) = \sqrt{\sum_i (s_i - t_i)^2} \tag{5.4}$$

$$D_{KL}(S,T) = \sum_i t_i log \frac{t_i}{s_i} \tag{5.5}$$

$$D_{JS}(S,T) = \frac{1}{2}\left[D_{KL}\left(S, \frac{S+T}{2}\right) + D_{KL}\left(T, \frac{S+T}{2}\right)\right] \tag{5.6}$$

$$D_\alpha(S,T) = \frac{1}{\alpha - 1} log\left(\sum_i s_i^{(1-\alpha)} t_i^\alpha\right), \alpha \geq 0 \tag{5.7}$$

Table 5.2 shows correlations between the similarity measures (5.1)-(5.7) and the accuracies given by the two classifiers, linear SVM and VP. Domain similarity and cross-domain accuracies are calculated on 42 source-target domain pairs from our seven domain dataset. In contrast to the outcomes of Asch and Daelemans (2010), Renyi divergence with $\alpha = 0.99$ gives the worst result. In general, most similarity measures, except for Renyi divergence and L_1, have a high correlation with the two classifiers and the difference between them is not statistically significant. The best results on average are delivered

measure	SVM	VP	average
cosine	0.886	0.897	0.892
$-L_1$	0.787	0.808	0.798
$-L_2$	0.888	0.900	0.894
$-\chi^2$	0.892	0.906	0.899
$-D_{KL}$	0.899	0.880	0.890
$-D_{JS}$	0.894	0.908	0.901
$-D_{0.99}$	0.780	0.658	0.719

Table 5.2: Correlation between various domain similarity measures and the cross-domain accuracies of SVM and VP calculated on 42 source-target domain pairs.

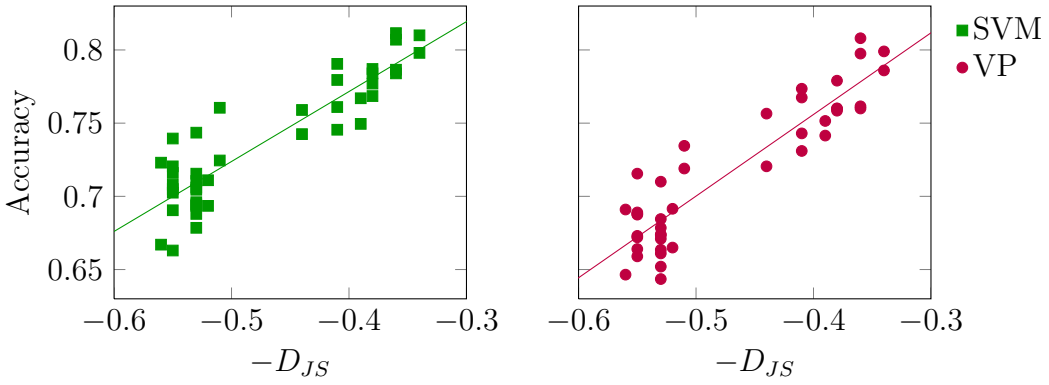

Figure 5.2: The relationship between $-D_{JS}$ values and the cross-domain accuracies of SVM and VP, calculated on 42 source-target domain pairs.

by χ^2 and D_{JS}, which is in accordance with Kilgarriff (2001) and Plank and van Noord (2011).

Figure 5.2 illustrates the relationship between $-D_{JS}$ and the cross-domain accuracies of the SVM and VP classifiers. We can clearly identify a boundary between domain pairs with low and high divergences, which lies between $D_{JS} = 0.4$ and $D_{JS} = 0.5$[7]. If we consider BO, DV and MU to form

[7]This result would benefit from a more formal proof with larger number of domains and the use of cross-validation. This will be a focus of our future work.

one group and TO, EL, KI and HE to form another group, then domain pairs with high divergences belong to the opposite groups while domain pairs with low divergences belong to the same group. This is in line with our observation about the similarity of BO, DV and MU on one side and TO, EL, KI and HE, on the other. Interestingly, the correlation between D_{JS} and the cross-domain accuracies is very strong for similar domains but it almost disappears for distinct domains. This means that for dissimilar domains D_{JS} can signal a substantial loss of accuracy but it cannot be used to predict its exact value.

5.3 Human annotation experiment

Product reviews are a good source of free labelled data for sentiment analysis and are widely used by many researchers in the field. It is usual to assume that consumer intentions expressed by their ratings coincide with reader perceptions after reading the reviews and, therefore, that all errors are due to the imperfection of the classification model. However, recent studies have demonstrated that the content of reviews does not always fully reflect the ratings assigned (Carrillo de Albornoz et al., 2011; Maks and Vossen, 2013). One possible reason for such a mismatch is that reviews and their ratings may express different aspects of the customer experience (Maks and Vossen, 2013). For example, a reviewer can rate a product highly but warn in the text about its negative aspects without mentioning its positive qualities since they are already expressed by the rating. Another explanation is the presence of "user bias", which means that ratings are not only related to review texts but

are also dependent on their authors (Li et al., 2011). Different reviewers can have different styles of writing and sentiment expression preferences informed by their personality, education, social interactions, etc. For example, the word "good" can mean "excellent" for some customers and "mediocre" for others. Therefore, review ratings can be quite inconsistent, which makes them difficult to judge even for humans.

Figures 3.13-3.17 show a decrease in accuracy of 20-25% when augmenting the binary sentiment scale with two more classes. This outcome is expected due to the increased complexity of the sentiment classification task. However, we suspect that the inconsistency of review ratings could also contribute to the performance loss. In this section, we describe an annotation experiment where a subset of the reviews was given to three native English speakers for manual labelling. The inter-annotator and annotator-reviewer agreements were then analysed to assess the task complexity and the quality of the review data.

5.3.1 Experiment description

The experiment consists of the manual annotation of a small subset of user reviews from four domains: BO, DV, EL and KI. The diversity of domains aims to reveal whether BO and DV are harder for humans to judge, as they are for automatic methods (see Section 3.5). This task was completed by three native English speakers, from here on referred to as coders or annotators (C_1, C_2 and C_3). The data sample consists of 100 reviews on each of the above

four product types (400 documents in total) from our initial corpus described in Section 3.1. Documents were selected randomly with the only condition of having an equal amount of reviews for each sentiment class. This resulted in a sample of 25 reviews for each number of stars (1*, 2*, 4* and 5*) per domain. However, the information that the data is balanced was hidden from the annotators.

The coders were given a .txt file with 400 reviews listed one by one in the following order: BO, EL, KI, DV. The texts were grouped by product type, and the product type itself was clearly stated at the beginning of the group. Only the actual texts of the reviews were given to the coders and no other additional information, such as product name, review title or reviewer name was shown to them. The coders were told that all reviews are rated as 1*, 2*, 4* or 5* and were requested to make their judgements on that rating. In case of uncertainty, an option to use the 3* rating was given. The coding instructions used in this experiment are shown in Appendix B.

The high subjectivity of the annotation task and its dependency on a coder's personality is indicated by the distribution of the coder judgements over the 5* scale (Table 5.3). For example, C_3 prefers to assign strong scores compared to C_2, who is more inclined to moderate scores. C_1 shows a tendency to give lower ratings than those assigned by the reviewers.

The analysis of the most disputed annotations, when all three coders disagree or one or more of them gives an opposite sentiment to the actual rating, reveals the following ambiguous contexts:

- Comparison between two different products without a clear indication of which product is under review.

- Equal amount of praise and criticism, making it difficult to judge which is more prominent for the reviewer.

These cases were identified during our preliminary data analysis and stated in the revised coding instructions for use in uncertain situations (Appendix B). However, generally the coders were able to give a definite judgment (see Table 5.3).

Coders	1*	2*	3*	4*	5*
C_1	113	94	6	98	89
C_2	98	108	2	107	85
C_3	123	75	0	83	119

Table 5.3: Distribution of the coder judgements over the 5* scale.

5.3.2 Evaluation

The usual practice for a data annotation task, which we employ, is to combine judgements provided by different coders to obtain a gold standard annotation (C^*). This was achieved using majority and uncertainty rules. We applied the majority rule when two or more coders agree on a rating. When all three judgements are different, the uncertainty rule, which always assigns 3* to the final rating, is used.

We applied two widely-exploited metrics for measuring agreement between the coders and reviewers:

- Percentage agreement - the percentage of judgements on which two coders agree (Scott, 1955).

- κ-coefficient - the percentage of agreement corrected for the agreement expected by chance (Cohen, 1960).

All agreements are calculated for both multiclass (4 review ratings + uncertain) and binary (positive, negative, uncertain)[8] classification. To analyse whether some domains are more challenging for sentiment annotation, the agreement per domain is also computed.

The interpretation of κ "is still little more than a black art" (Artstein and Poesio, 2008, page 576). NLP researchers usually follow conventions reported in the work of Carletta (1996) that $\kappa > 0.8$ refers to a reliable annotation while $0.67 < \kappa < 0.8$ allows "tentative conclusions to be drawn" (Carletta, 1996, page 252). The authors of more recent study (Poesio and Artstein, 2005) claim that only $\kappa > 0.8$ can ensure a good quality annotation.

Our evaluation addresses the objectives posed at the beginning of the section surrounding task complexity and human performance for the multiclass and binary cases, as well as inconsistency in the review ratings.

5.3.2.1 Task complexity

Task complexity can be estimated by inter-coder agreement between the pairs of coders C_1C_2, C_1C_3 and C_2C_3. Naturally, low agreement indicates

[8] Strictly speaking it is not a binary case because the coders could choose the uncertainty option. However, i) they used it very rarely (Table 5.3) and, ii) the 3rd class is a definite error since no such class exists in the data.

that the coders had problems annotating the data, which could be due to the high complexity of the task, confusing coding instructions, or both. Table 5.4 displays κ agreements between the pairs of coders. For multiclass classification none of the coder pairs reached $\kappa \geq 0.8$. Moreover, two coder pairs C_1C_3 and C_2C_3 did not surpass the threshold of 0.67. Binary classification is naturally easier and, accordingly, the overall κ agreement is around 0.9. This means that the task complexity increases significantly when two more sentiment classes are included.

	multiclass			binary		
Domain	C_1C_2	C_1C_3	C_2C_3	C_1C_2	C_1C_3	C_2C_3
BO	0.66	0.54	0.54	0.88	0.85	0.88
EL	0.66	0.54	0.57	0.85	0.88	0.79
KI	0.77	0.73	0.72	0.96	0.92	0.96
DV	0.71	0.69	0.71	0.98	0.92	0.94
All	0.70	0.62	0.63	0.92	0.89	0.89

Table 5.4: κ coefficients between coders C_1, C_2, C_3.

The κ agreement varies from one domain to another but its values are substantially higher for the two last domains (KI and DV) compared to the first domains (BO and EL). We think that this phenomenon is not only due to the higher difficulty of BO and EL for manual annotation but also because of the experience the coders gained while annotating the data. Therefore, the later annotations could be more reliable. To prove this hypothesis, a review sample could be randomised over the product types before conducting the annotation experiment; we leave this for future work. Since the pair C_1C_2

has the highest agreement, we consider the annotations provided by C_1 and C_2 to be the most reliable, while C_3 can be seen as an outlier.

5.3.2.2 Human performance

Human performance on the task is assessed by calculating percentage agreements and κ coefficients between the 4 annotations (given by the 3 coders and the gold standard) and reviewers (C_1U, C_2U and C_3U, C^*U) and is shown in Tables 5.5 and 5.6. We observe a similar picture to the inter-coder agreement reported above: in the multiclass case no coder reached $\kappa = 0.8$. Additionally, coders C_2, C_3 and gold standard C^* demonstrated very low agreement with the reviewers (Table 5.5). Interestingly, C^* shows lower agreement than C_1 which proves that low κ is due to the task complexity rather than incomplete or confusing coding guidelines. There are slight differences in the κ agreement from one domain to another, and κ for KI is consistently higher for all coders. Likewise κ for *BO* is consistently lower for all coders in comparison to other domains. We expected DV reviews to be more complex for manual annotation than EL reviews due to their similarity with BO and KI respectively. However, the evaluation revealed the opposite: κ agreement is higher for DV than for EL although the difference is not significant. This may be an effect of the coder experience acquisition mentioned in the previous section.

Percentage agreement corresponds to the accuracy that a human annotator achieves on the dataset. Table 5.6 shows that C_1 obtained

	multiclass				binary			
Domain	C_1U	C_2U	C_3U	C^*U	C_1U	C_2U	C_3U	C^*U
BO	0.63	0.49	0.41	0.52	0.90	0.86	0.82	0.88
EL	0.67	0.52	0.53	0.58	0.84	0.80	0.84	0.84
KI	0.76	0.61	0.67	0.71	0.96	0.92	0.92	0.92
DV	0.68	0.55	0.59	0.65	0.88	0.90	0.88	0.90
All	0.69	0.55	0.55	0.62	0.90	0.87	0.87	0.89

Table 5.5: κ coefficients between coders C_1, C_2, C_3, gold standard C^* and reviewers U.

	multiclass				binary			
Domain	C_1U	C_2U	C_3U	C^*U	C_1U	C_2U	C_3U	C^*U
BO	72	62	56	64	95	93	91	94
EL	75	64	65	68	92	90	92	92
KI	82	71	75	78	98	96	96	96
DV	76	66	69	74	94	95	94	95
All	76	66	66	71	95	94	93	94

Table 5.6: Percentage agreements between coders C_1, C_2, C_3, gold standard C^* and reviewers U.

the highest accuracy of 76%. Since C_1 achieved the maximum agreement with the reviewers we use the C_1 accuracies as a reference for the human performance. Comparing the best human performance with the machine learning upper bound (Table 3.4), the differences between them are relatively moderate and are around 10-16%: 72% vs. 62% for BO, 75% vs. 64% for EL, 82% vs. 66% for KI and 76% vs. 65% for DV[9]. Moreover, coders C_2 and C_3 performed 10% worse on average than C_1; therefore, compared to their efforts, the difference between the human and machine accuracies is even less pronounced.

[9] The direct comparison is not completely fair since the sample for manual annotation is only a small fraction of our data. Nevertheless, it can give us an approximate idea about the accuracy levels for humans and automatic algorithms.

5.3.2.3 Inconsistency in review ratings

As mentioned above, several studies have indicated that review data can be noisy due to the mismatch between review ratings and their corresponding texts (referred to here as label mismatch). To explore this, we compare inter-coder and coder-reviewer agreements, assuming that higher inter-coder agreement casts doubt on the conformity of the review texts and ratings. Since C_3 seems to be an outlier we only compare κ agreements C_1C_2, C_1U and C_2U. Tables 5.4 and 5.5 show that the inter-coder agreement C_1C_2 is slightly better than the coder-reviewer agreements C_1U and C_2U, where the difference between them is higher for C_2U and almost disappears for C_1U. However, this difference is too small to prove label mismatches in our review data. To find stronger evidence, we suggest another approach: identifying reviews on which the two most reliable coders C_1 and C_2 agreed but where their judgement is different from the review rating. Using this approach, we detected 68 reviews satisfying these conditions, which means that at least 17% of reviews in our data sample do not conform to their ratings. Some examples of label mismatches identified in our data are displayed in Table 5.7. Therefore, although task complexity increases considerably for multiclass classification, a percentage of errors is due to inconsistencies in the labelled data. The human annotation experiment suggests a rough approximation to the machine learning upper bound lying between 76%, as shown by the most reliable coder, and 83%, which excludes evident cases of label mismatches.

N	Review texts	author rating	coder rating
1	Within 2 weeks, the pause & brew didn't exist. The coffee poured all over when I tried to pull the carafe for a cup. My attempts at fixing it were for naught. What a waste of money! I'm not one who returns items. I just won't buy Braun again!	2	1
2	like the other guy, not much can be said. works. works well. end of story. The only thing i would have liked to see was integration of sorts into a pci slot or something, to keep it within the case	5	4
3	I wish that I had purchased the sheet set earlier so that I could have enjoyed them all winter, they are so comfortable, even with flannel pj's I didn't have any problem turning over. I completely recommend this sheet.	4	5

Table 5.7: Examples of label mismatches in product reviews.

5.4 Summary

In this chapter, we established the data characteristics which affect semi-supervised and cross-domain sentiment classification. First, we introduced the domain complexity concept, which reflects the difficulty for machine learning algorithms to learn from data. We estimated domain complexity by vocabulary richness and established a high correlation between in-domain sentiment classification accuracies and such vocabulary richness measures as TTR and the percentage of rare words. Second, the dependence of the cross-domain classification results on source-target domain pairs revealed another important data characteristic, called domain similarity. Following other cross-domain studies, various similarity functions were assessed using

their Pearson correlation with the cross-domain accuracies. The χ^2 and D_{JS} functions had a marginal advantage over other measures. Finally, we conducted a manual annotation of a subset of reviews and detected label mismatches between review texts and ratings. Therefore, we conclude that some machine learning errors are due to inconsistencies in the review ratings.

Chapter 6

Semi-supervised experiments

In this chapter, the evaluation of our graph-based sentiment analysis system in semi-supervised settings is carried out. In particular, the performance of four graph-based algorithms, LP, LP_{γ}, $LP_{\alpha\beta}$ and $RANK$, introduced in Chapter 4 is examined. The evaluation experiments aim to answer the following questions:

1. Which modifications to the graph-based model improve algorithm performance and which algorithm delivers the best results?

2. Do graph-based approaches benefit from normalisation and which normalisation technique is most effective?

3. Does the hierarchical probability combination rule have an advantage over the maximum probability rule?

4. How much in-domain labelled data is needed to match the accuracy of fully supervised classification?

5. Does domain complexity influence the graph-based results?

6. What are the optimal parameter values for the graph algorithms and are the algorithms sensitive to variations in their parameters?

7. Does the hybrid similarity metric perform best when tested extrinsically?

8. Does adapting the SO-CAL dictionaries to the review data improve accuracy?

9. How do graph-based approaches perform in comparison to various state-of-the-art semi-supervised methods?

The chapter is organised as follows: Section 6.1 describes the semi-supervised evaluation setup and all the subsequent sections address the questions stated above. In particular, Section 6.2 addresses question 1, Section 6.3 questions 2-5, Section 6.4 question 6, Section 6.5 question 7, Section 6.6 question 8 and Section 6.7 question 9. The evaluation results are summarised in Section 6.8.

6.1 Experimental setup

The experiments were carried out separately for each domain. We randomly divided our data into 5 folds, where one was used for parameter tuning and 4 for testing the algorithms in the cross-validation setup. Thus, for each fold, 400 examples were used for testing/tuning and the remaining 1600 instances were split into labelled and unlabelled sets. We gradually increased the amount of labelled data from 50 to 800 examples to analyse the impact of the labelled data size on the performance of the algorithms.

During the tuning stage, we adjusted the values of the following parameters: α, β, the number of unlabelled neighbours k_u and the proportion of labelled neighbours with respect to the labelled data size Δ_l. We used Δ_l instead of k_l as we found it more natural for the variable sizes of labelled data. The parameter search was run over the following ranges:

- $\alpha \in \{1, 2, 5, 10, 50, 100, 200\}$,
- $\beta \in \{0.2, 0.5, 1, 2, 5\}$,
- $k_u \in \{5, 10, 20, 50, 100, 200\}$,
- $\Delta_l \in \{0.05, 0.1, 0.2, 0.3, 0.4, 0.5\}$.

Our selection of optimal parameter values was based on two criteria: maximisation of the mean F-score, $\bar{F}_1$, averaged over all domains and labelled data sizes, and minimisation of the $\bar{F}_1$ variance over domains. This ensures both accurate and balanced performance over all domains. $LP_{\alpha\beta}$ also requires initial approximations for labels, which we obtained by applying our baseline classifier trained on the corresponding fraction of labelled data. Parameter values were tuned separately for each LP variant and algorithm configuration. By algorithm configuration we mean which normalisation technique, if any, and which probability combination rule were used at the post-processing stage. The configuration with the maximum probability combination rule and no normalisation is referred to as the basic configuration.

6.2 LP and its modifications in the basic configuration

First, LP and its variants are compared in the basic configuration. This automatically excludes $RANK$ as it uses CMN after each iteration. Figure 6.1 presents the accuracies averaged over domains given by LP, LP_γ and $LP_{\alpha\beta}$ for binary sentiment classification. We also give the semi-supervised baseline (B-line) (Section 3.5.1) and the upper bound (U-bound), which corresponds to the best result of fully supervised classification (Table 3.4). The baseline and upper bound are both averaged over domains.

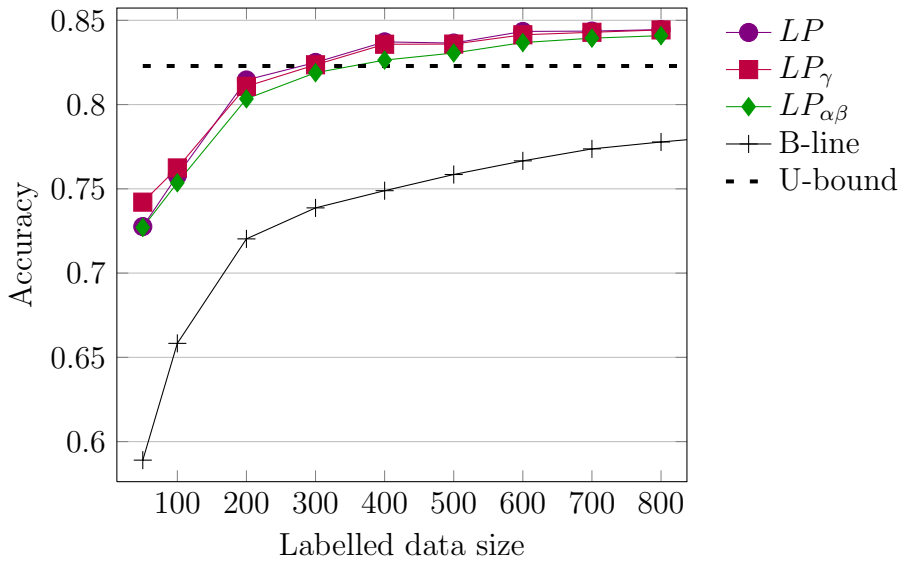

Figure 6.1: Best accuracy averaged by domain for LP, LP_γ and $LP_{\alpha\beta}$ (binary case).

LP and LP_γ demonstrate almost identical results, which casts doubt on the positive contribution of the parameter γ. Both algorithms show a slight

advantage over $LP_{\alpha\beta}$, which indicate that the initial predictions are not very helpful. At the same time, all graph-based results are considerably higher than the baseline accuracies. This difference is largest (about 15 ppt) when less labelled data is available and it decreases to 6 ppt when 800 labelled examples are used. In addition, the graph-based accuracies reach the upper bound with only 300 examples and continue increasing gradually when more labelled data is added.

For the multiclass case, the $\bar{F}_1$ values averaged over domains are reported as we want both the accuracy and the macroaveraged F-score to be high (Figure 6.2). Unlike in the binary case, $LP_{\alpha\beta}$ outperforms the other two methods, although this difference is not statistically significant. In general, the graph-based multiclass results do not show a clear advantage over the baseline as the binary accuracies do, and they do not reach the upper bound even for the maximum number of labelled examples. However, in contrast to the binary case, the parameter γ gives a small contribution in classification performance. This determines our preference for the more complex algorithm LP_γ over LP for further experiments.

Overall, the graph-based algorithms in the basic configuration are found to be more successful for binary sentiment classification than for multiclass classification. We think that the main reason for this is the limitations of the similarity metric, which better estimates the simpler case of two sentiment classes. At the same time, as shown in the following section, normalisation significantly improves the multiclass results (Section 6.3). Therefore, another

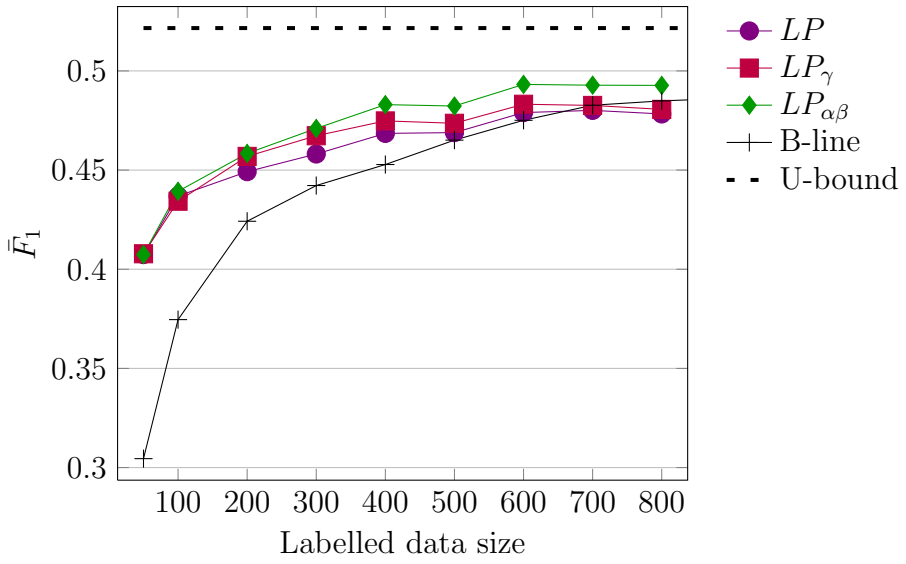

Figure 6.2: Best ($\bar{F}_1$) averaged over domains for LP, LP_γ and $LP_{\alpha\beta}$ (multiclass case).

explanation for the poor multiclass performance in the basic configuration could be the unbalanced distribution of sentiment classes.

6.3 The impact of normalisation and the hierarchical probability combination rule

In this section, we investigate how different algorithm configurations influence the performance of graph-based algorithms. The analysis is conducted for all LP variants excluding LP itself as the results it showed were worse or identical to LP_γ. For the binary case, LP_γ and $LP_{\alpha\beta}$ have three configurations: basic, CMN and LB. The last two configurations use either CMN or LB to normalise the output labels at the post-processing stage. $RANK$ has only two configurations since its basic configuration coincides

with CMN. As in the previous section, the algorithms are compared by their accuracies averaged over domains when the amount of labelled data is fixed.

For the multiclass case, we consider two more configurations: $HIER$, corresponding to the hierarchical probability combination rule used at the post-processing stage, and $HIER{+}LB$ which applies LB in addition to $HIER$. Consequently, LP_γ and $LP_{\alpha\beta}$ have five configurations, while $RANK$ has four. As evaluation metrics, we use accuracy, $macroF_1$, as well as their mean $\bar{F}_1$. Unless otherwise specified, the evaluation metrics are averaged over domains.

6.3.1 The binary case

Both CMN and LB significantly improve the performance of LP_γ and $LP_{\alpha\beta}$ in the basic configuration (Figure 6.3). The CMN and LB configurations also demonstrate similar behaviour, which could be because the binary data sets are balanced. LB does not improve the $RANK$ performance, therefore, we do not consider it in our subsequent experiments.

In Figure 6.4, the most successful configurations of LP_γ, $LP_{\alpha\beta}$ and $RANK$ together with the upper bound of accuracy are compared. $LP_\gamma{+}LB$ clearly outperforms the other algorithms. Moreover, it achieves considerably better results with a relatively small labelled data size. For example, the accuracy averaged over domains surpasses the boundary of 0.8 with just

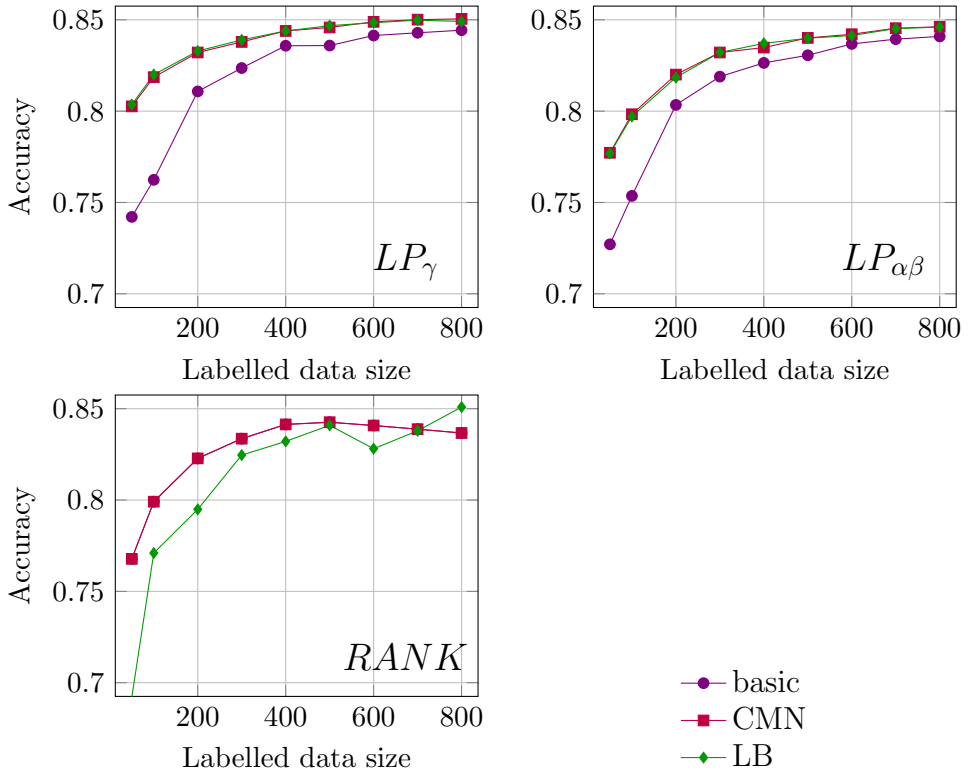

Figure 6.3: Best accuracy averaged over domains for different algorithms and different normalisation techniques (binary case).

50 labelled documents. When 100 labelled examples are given, the average accuracy approaches the upper bound level.

In Figure 6.5, the accuracies and MSE values of LP_γ+LB for individual domains are reported. There is a strong dependency of the graph-based results on domain complexity, so that more complex domains have significantly lower accuracy and higher MSE values. However, even for more complex domains, 100-200 labelled examples are enough to reach accuracies above 0.8 and MSE values below 0.2. Simpler domains yield similar results with only 50 labelled documents. In general, the MU domain has the lowest

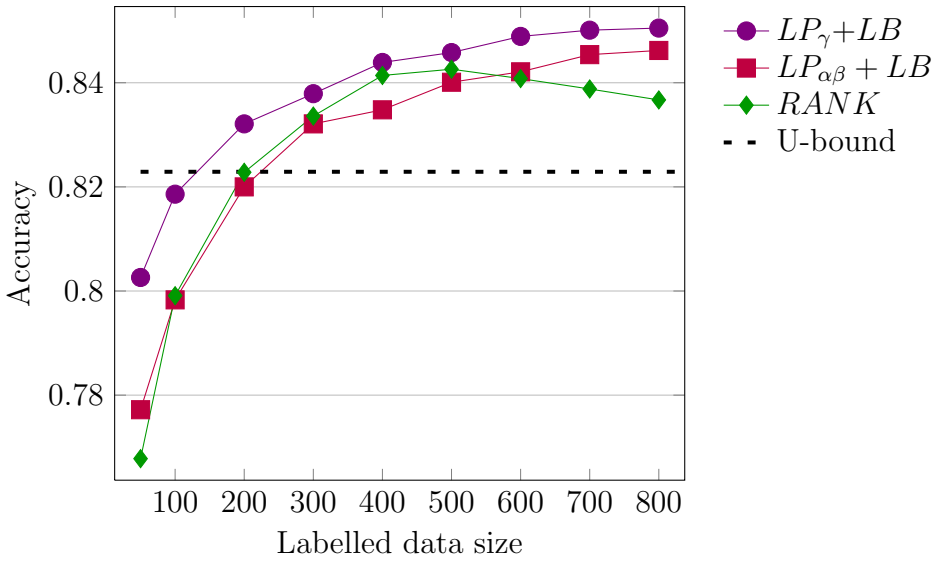

Figure 6.4: Best accuracy averaged over domains for the most successful algorithms and normalisation techniques (binary case).

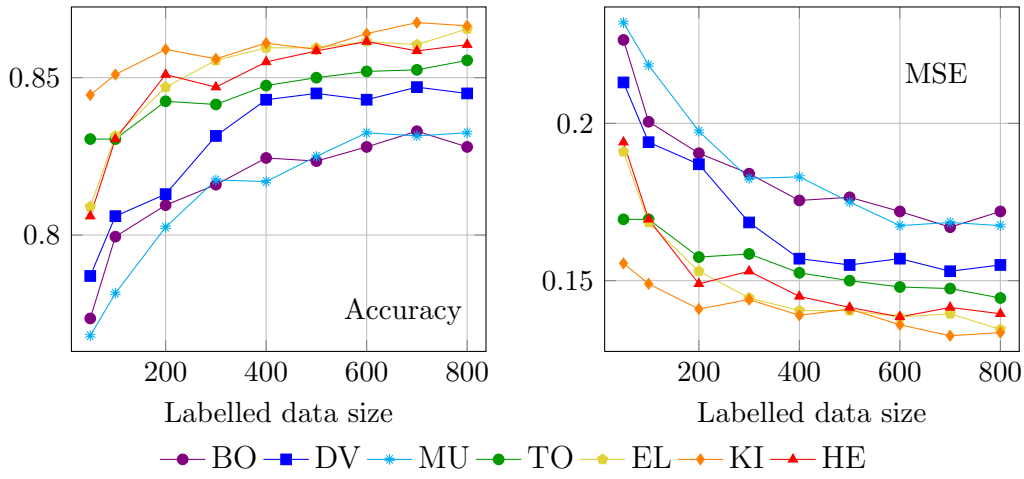

Figure 6.5: Accuracy and MSE obtained with $LP_\gamma + LB$ for each domain (binary case).

values, which is in agreement with our conclusions about it having the highest level of complexity.

6.3.2 The multiclass case

The multiclass classification results for different algorithms and their configurations are given in Figure 6.6. To evaluate each algorithm in more detail, both accuracies and $macroF_1$ values are reported. The accuracies are displayed on the left of Figure 6.6 and the $macroF_1$ values are on its right.

The accuracies of all graph-based algorithms are substantially higher than the baseline. Some configurations surpass the upper bound of accuracy when the number of labelled documents is sufficient (> 400). The $HIER$ configuration is generally the most successful for all algorithms, while LB is the least successful. Comparing the three algorithms, $RANK$ in basic and $HIER$ configurations achieves the highest accuracy for all sizes of labelled data. Although LP_γ yields overall lower accuracies than $LP_{\alpha\beta}$ for all configurations, it has a slight advantage when few labelled examples are available. Indeed, LP_γ+LB reaches accuracy levels comparable to $RANK$ for 50-100 labelled examples.

A completely different picture can be observed when analysing $macroF_1$ (Figure 6.6). First, there are substantial differences between the $macroF_1$ values corresponding to different algorithm configurations. As a rule, $macroF_1$ is very high for the LB and $HIER$+LB configurations, while it drops drastically for the basic, CMN and $HIER$ configurations. In this respect, $RANK$ is the most balanced. Indeed, $RANK$ and $RANK$+$HIER$ achieve the highest accuracy and, at the same time, their $macroF_1$ is

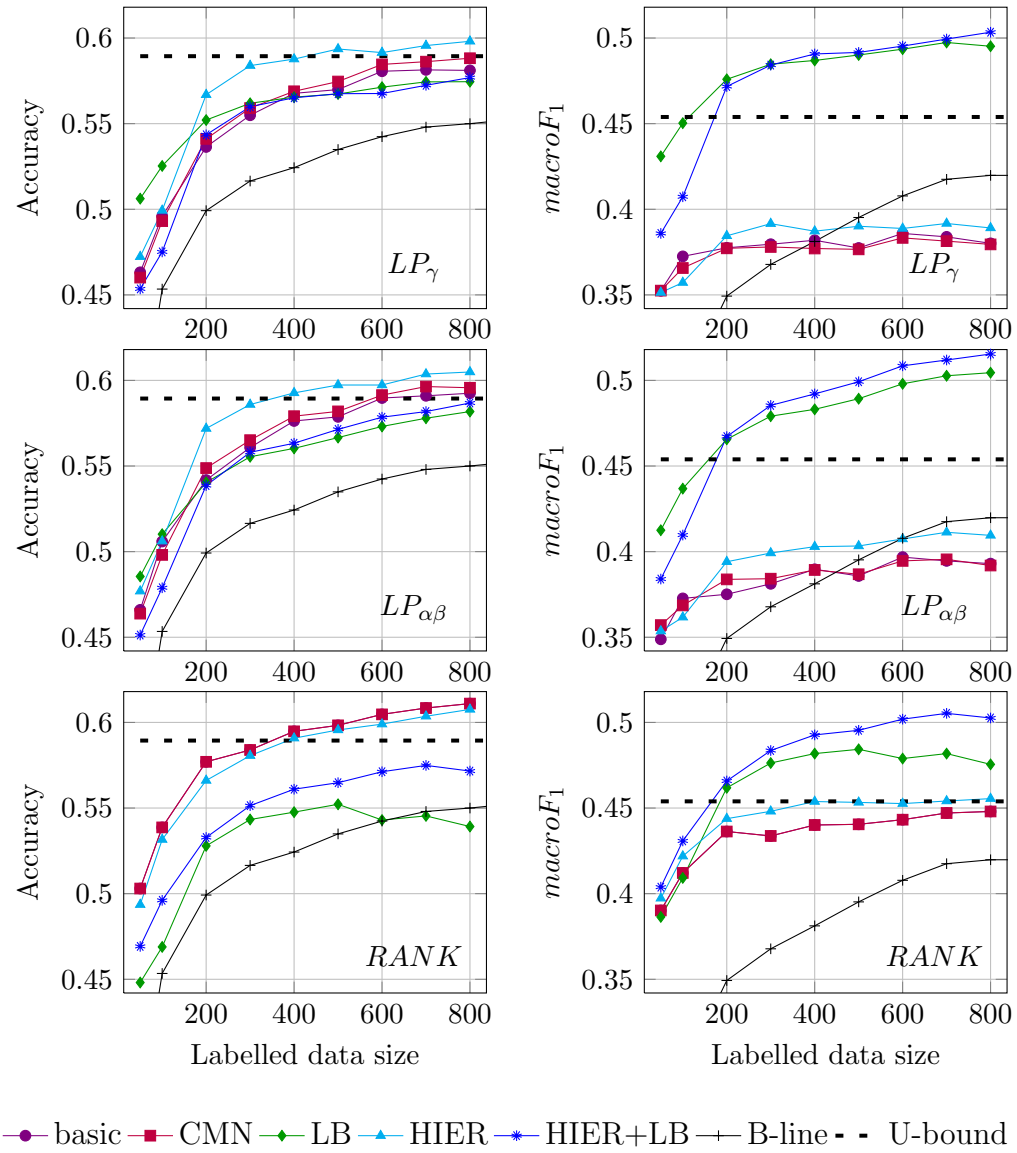

Figure 6.6: Accuracy and $macroF_1$ averaged over domains for LP_γ, $LP_{\alpha\beta}$ and $RANK$ in different configurations (multiclass case).

reasonably high, reaching the upper bound with 300 labelled documents. The basic, CMN and $HIER$ configurations of LP_γ and $LP_{\alpha\beta}$ have low $macroF_1$ values while yielding a high accuracy. This suggests that although these configurations correctly classify a large number of documents, the

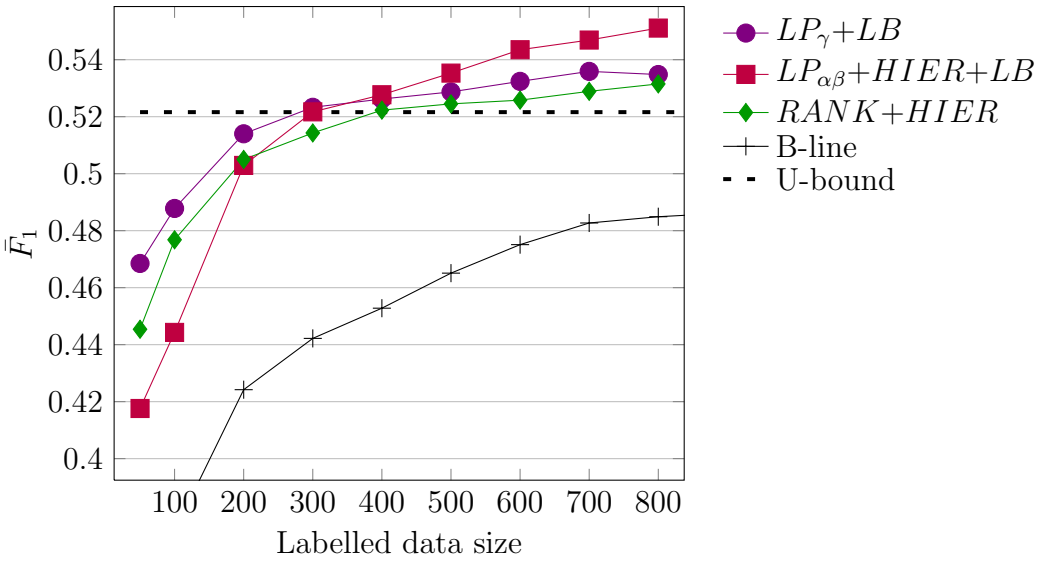

Figure 6.7: Best $\bar{F}_1$ averaged over domains for the most successful algorithms, normalisation techniques and probability combination rules (multiclass case).

majority of these documents belong to strongly positive and strongly negative sentiment classes as the most numerous in the data. Therefore, the basic, CMN and $HIER$ configurations are not very helpful for multiclass classification. In contrast, the LB and $HIER$+LB configurations of LP_γ and $LP_{\alpha\beta}$ are able to achieve a very high $macroF_1$, while maintaining accuracy levels comparable to other configurations. Overall, LP_γ+LB, $LP_{\alpha\beta}$+$HIER$+LB and $RANK$+$HIER$ give the highest trade-off between accuracy and $macroF_1$ (Figure 6.7). LP_γ+LB performs best for small amounts of labelled data and surpasses the upper bound with only 300 labelled examples. $LP_{\alpha\beta}$+$HIER$+LB achieves the most accurate results when the amount of labelled data is more than 300 examples. Finally, $RANK$+$HIER$ yields the highest accuracy and its $\bar{F}_1$ does not differ

significantly from the best performance delivered by LP_γ+LB. Therefore, this method should be preferred if accuracy is more important than the balanced representation of all sentiment classes in the final results.

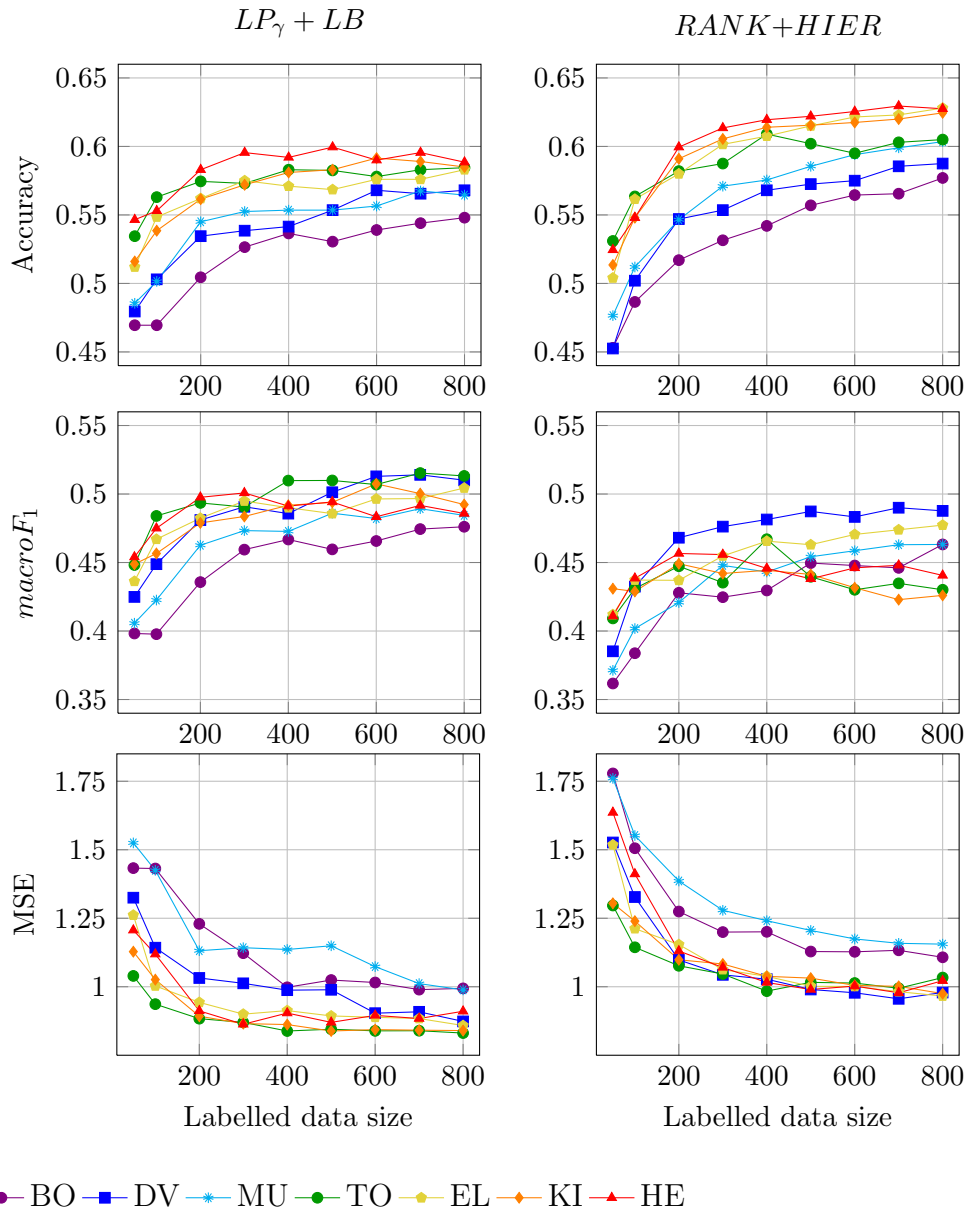

Figure 6.8: Accuracy, $macroF_1$ and MSE obtained with LP_γ+LB and $RANK$+$HIER$ for each domain (multiclass case).

To obtain deeper insights into the behaviour of the best graph-based algorithms we report their accuracy, $macroF_1$ and MSE separately for each domain (Figure 6.8). First of all, similarly to the binary case, the multiclass results are strongly correlated to domain complexity, which is especially clear for the accuracy and MSE values. Second, the LP_γ+LB graphs of all three evaluation functions are almost parallel to the X-axis when the number of labelled examples is higher than 200. This means that the algorithm does not benefit much from additional labelled data. At the same time, its performance for a small number of samples is very high. Moreover, LP_γ+LB surpasses $RANK$+$HIER$ using all evaluation metrics when the labelled data contains less than 100 examples. Thus, LP_γ+LB yields accurate results with a relatively small amount of human effort. Another advantage of LP_γ+LB over $RANK$+$HIER$ is its lower MSE, which indicates smaller differences between erroneous and actual sentiment scores. Since $RANK$+$HIER$ better classifies strongly positive and negative sentiment classes as the most numerous in the data, its MSE does not fall below 1 even for large amounts of labelled data. Generally, the graphs reveal that it is not necessary to have lots of labelled examples to achieve high performance. More labelled examples yield a very moderate gain and, in some cases, even make the results worse (see macroaveraged F-score for KI, HE and TO). Overall, 300-400 labelled documents are enough to outperform the upper bound levels (Figure 6.7).

6.3.3 Discussion

Our findings regarding the best algorithm configurations are similar for both binary and multiclass cases. If the $HIER$ configuration is ignored, as it is impossible in the binary case, we obtain the same set of the most successful algorithms for both tasks: LP_{γ}+LB, $LP_{\alpha\beta}$+LB and $RANK$. LP_{γ}+LB is proved to be the most effective overall. The excellent results of the LB configuration indicate that the similarity metric used in our experiments distinguishes well between different sentiment grades. Therefore, a priori knowledge about the proportion of documents in each class can substantially improve the final results. Interestingly, the more complex $LP_{\alpha\beta}$, which incorporates initial ratings given by external classifiers, performs worse overall than the simpler LP_{γ}. Figure 6.6 shows that $LP_{\alpha\beta}$ delivers poor results when a small amount of labelled data is available. For larger amounts of labelled data the $LP_{\alpha\beta}$ configurations usually outperform the same configurations of LP_{γ}, which could be due to the increased reliability of the initial predictions provided by external classifiers. In particular, $LP_{\alpha\beta}$+$HIER$+LB is more effective than LP_{γ}+LB when over 400 labelled examples are available (Figure 6.7). However, the fact that LP_{γ}+LB achieves accurate results with the least manual effort makes this graph-based algorithm much more valuable and useful.

6.4 Sensitivity to parameter variations

In the previous sections, we tuned the parameters of graph algorithms to find the combination of parameter values which delivered the best results. The optimal values obtained were then used to compare the algorithms in order to select the most effective method. However, the highest performance yielded for a set of parameters is not the only criterion for choosing the best algorithm. It is even more crucial to analyse stability of the algorithms when small variations to parameter values are introduced. Since optimal values might be dependent on data, classification task and experimental setup, stability is an important property which ensures that the results obtained are close to an algorithm's best performance. In this section, we explore the sensitivity of the most successful graph-based algorithms to variations in their parameters. As binary classification is a special case of multiclass classification, we conduct our analysis for the multiclass case only.

In Table 6.1, the highest and lowest values $\bar{F}_1$ are reported for all LP variants and their configurations when the parameters of the algorithms vary within a certain range. To establish whether there is a dependency between the algorithms' stability and the number of labelled examples, three sizes of labelled data are used: 100, 300 and 700 examples. The following observations can be made:

- The LB configuration always yields relatively stable results. Moreover, the stability increases with increasing labelled data size. The difference

between its minimum and maximum values varies from 2-4 ppt for 700 labelled examples to 5-8 ppt for 100 examples. In contrast, the performance of the basic configuration can drop up to 23 ppt from its best to its worst result. The relatively low performance variations of the LB configuration imply that the instability of other configurations is mainly due to the incorrect final class distribution.

- The minimum $\bar{F}_1$ for all LB configurations is higher than the baseline. Moreover, it matches the upper bound when the labelled data size is 700 documents.
- $LP_{\alpha\beta}$ is more stable than LP_{γ}, which becomes more obvious for larger amounts of labelled data. However, LP_{γ} is more likely to achieve better results. This is due to the regularisation term in equation (4.11) (Section 4.5.2), which does not allow the final results of $LP_{\alpha\beta}$ to differ much from the initial values.
- All $RANK$ configurations are quite stable with respect to parameter variations, which suggests that applying CMN after each iteration prevents a highly skewed final class distribution. However, more labelled data is not always helpful for $RANK$. For example, the basic and $HIER$ configurations show a significant drop in performance for some parameter values. This phenomenon is discussed further when we analyse in more detail the effect of parameter variations on the $RANK$+$HIER$ behaviour.

The optimal parameter values are dependent on the algorithm and its

Method	Configuration	Labelled data size					
		100		300		700	
		$\bar{F}_1$min	$\bar{F}_1$max	$\bar{F}_1$min	$\bar{F}_1$max	$\bar{F}_1$min	$\bar{F}_1$max
LP_γ	basic	0.25	0.48	0.25	0.47	0.31	0.50
	CMN	0.30	0.45	0.27	0.47	0.37	0.49
	LB	**0.42**	0.50	**0.47**	0.53	**0.51**	0.55
	HIER	0.21	0.43	0.36	0.49	0.40	0.50
	HIER+LB	0.25	0.44	0.37	0.54	0.46	0.55
$LP_{\alpha\beta}$	basic	0.27	0.45	0.29	0.47	0.38	0.50
	CMN	0.29	0.45	0.34	0.48	0.43	0.50
	LB	**0.42**	0.47	**0.47**	0.52	**0.52**	0.54
	HIER	0.22	0.43	**0.44**	0.49	0.47	0.52
	HIER+LB	0.24	0.45	**0.47**	0.52	**0.52**	0.55
RANK	basic	**0.40**	0.47	**0.45**	0.51	0.35	0.52
	LB	**0.40**	0.48	**0.45**	0.52	**0.49**	0.53
	HIER	**0.40**	0.47	**0.46**	0.52	0.41	0.53
	HIER+LB	**0.40**	0.48	**0.46**	0.52	**0.50**	0.54
B-line		0.37		0.44		0.48	
U-bound		0.52		0.52		0.52	

Table 6.1: Variations of $\bar{F}_1$ for LP_γ, $LP_{\alpha\beta}$ and $RANK$, different sizes of labelled data (100, 300 and 700 examples) and parameter ranges: α=200, $\beta \in \{0.2, 0.5, 1, 2, 5\}$, $k_u \in \{5, 10, 20, 50, 100\}$ and $\Delta_l \in \{0.05, 0.1, 0.2, 0.3, 0.4, 0.5\}$ (minimum values that surpass the B-line are highlighted, minimum values that surpass the U-bound are underlined).

configuration (Table 6.2). As a rule, the *LB* configuration requires higher values of β, k_u and Δ_l indicating that the best results are due to a greater impact of unlabelled data and a higher number of labelled and unlabelled neighbours. In contrast, the basic and *CMN* configurations give better results with lower β, k_u and Δ_l values, which means that they rely more on labelled data and closely related labelled and unlabelled neighbours. Therefore, large-scale averaging achieves more accurate relative sentiment

Method	Configuration	Parameters			
		β	k_u	Δ_l	α
LP_γ	basic	0.2	5	0.1	—
	CMN	0.2	5	0.1	—
	LB	5	50	0.5	—
	$HIER$	0.2	5	0.3	—
	$HIER$+LB	0.2	5	0.4	—
$LP_{\alpha\beta}$	basic	0.2	5	0.1	200
	CMN	0.2	5	0.1	200
	LB	1	10	0.3	200
	$HIER$	0.2	5	0.3	200
	$HIER$+LB	0.2	5	0.3	200
$RANK$	basic	0.2	100	0.5	—
	LB	1	100	0.5	—
	$HIER$	0.2	100	0.5	—
	$HIER$+LB	1	100	0.5	—

Table 6.2: Optimal parameter values for different configurations of LP_γ, $LP_{\alpha\beta}$ and $RANK$.

scores, while small-scale averaging helps to obtain correct absolute sentiment values.

We now explore in more detail sensitivity to parameter variations of the three algorithms, LP_γ+LB, $LP_{\alpha\beta}$+$HIER$+LB and $RANK$+$HIER$, as the most successful according to our experiments. Figure 6.9 displays the $\bar{F}_1$ results for variations of one of the parameters while the other parameters are left fixed at their optimal values. To understand whether sensitivity to parameter variations is affected by the labelled data size, the algorithms' performance is computed for different numbers of labelled examples (100, 300 and 700). Due to the poor performance of $LP_{\alpha\beta}$+$HIER$+LB for small amounts of labelled data, its sensitivity is assessed for only 300 and 700 examples. According to Figure 6.9, LP_γ+LB and $LP_{\alpha\beta}$+$HIER$+LB are

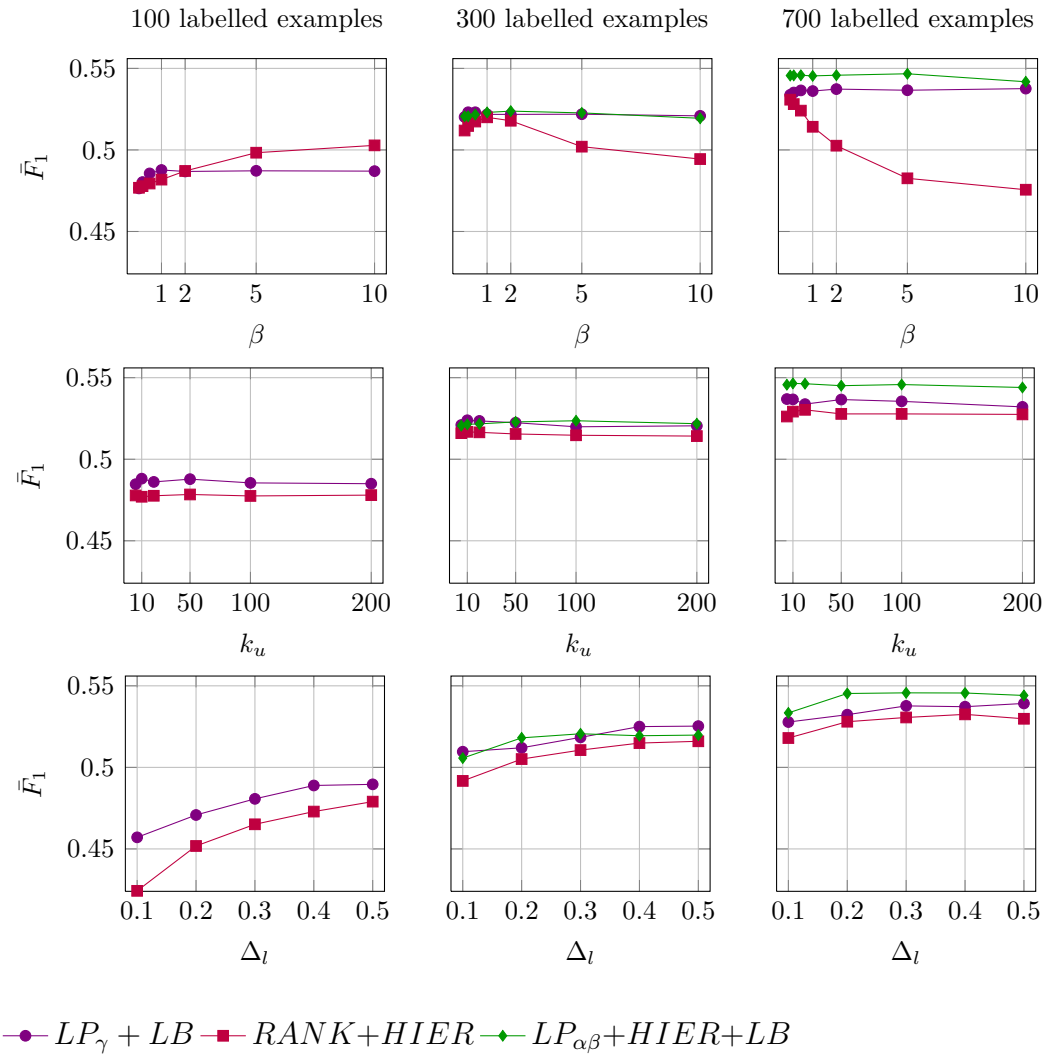

Figure 6.9: Sensitivity of LP_γ+LB, $LP_{\alpha\beta}$+$HIER$+LB and $RANK$+$HIER$ to variations of their parameters. The results are given for different sizes of labelled data: 100, 300 and 700 examples.

almost insensitive to variations of β and k_u. LP_γ+LB shows a small drop in performance for low values of β when 100 labelled examples are used. However, this drop is not statistically significant and the choice of $\beta \geq 1$ gives stable results for all labelled data sizes. $LP_{\alpha\beta}$+$HIER$+LB demonstrates

similar behaviour, with the difference being that its performance slightly decreases when $\beta = 10$ and, therefore, values of $\beta \leq 5$ are preferable. Concerning the parameter k_u there is a small downward trend with the increase of k_u for both LP_γ+LB and $LP_{\alpha\beta}$+$HIER$+LB. Thus, the choice of $k_u \leq 50$ ensures that the algorithms are more effective.

Unlike LP_γ+LB and $LP_{\alpha\beta}$+$HIER$+LB, $RANK$+$HIER$ is sensitive to variations of β. Moreover, the optimal values of β are dependent on the number of labelled documents. Higher values of β are beneficial for smaller sizes of labelled data, but have a negative effect when more labelled data is involved. Sensitivity of $RANK$+$HIER$ to variations of β increases with the amount of labelled data, which can be seen by a sharper decrease in performance. This is in agreement with the results of Table 6.1. $0.2 \geq \beta \leq 0.5$ yields the best overall results for all data sizes. This finding coincides with those of Wu et al. (2009), where the optimal γ was found to be 0.7, which corresponds to $\beta \approx 0.4$ in our experiments.

All three algorithms are sensitive to variations of Δ_l (Figure 6.9) having a strong preference for higher levels of Δ_l. When fewer labelled examples are used, higher values of Δ_l significant improve performance. For increased amounts of labelled data, the Δ_l graphs flatten and the performance becomes less dependent on variations of Δ_l. It is worth pointing out that Figure 6.9 justifies our choice of the parameter Δ_l instead of k_l. The optimal number of labelled neighbours depends on the number of labelled examples: ≈50 for 100 examples, ≈100 for 300 examples, and ≈200 for 700 examples. Although this

proportion does not stay the same, choosing a higher number of neighbours usually does not harm the performance. In contrast, restricting k_u to 50 stops the algorithms from achieving their best results for large amounts of labelled data. Overall, $\Delta_l = 0.5$ guarantees a high performance for all labelled data sizes.

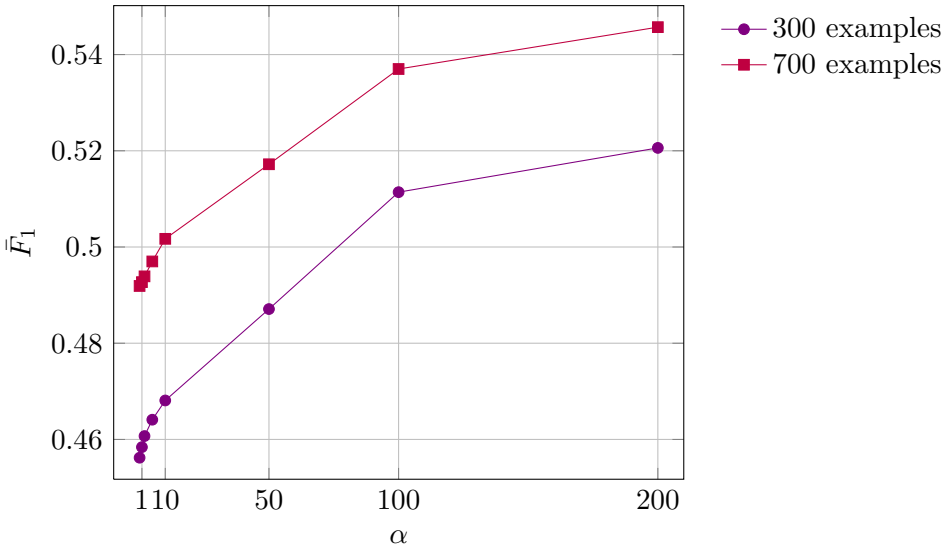

Figure 6.10: Sensitivity of $LP_{\alpha\beta}$+$HIER$+LB to variations of the parameter α. The results are given for 300 and 700 labelled examples.

$LP_{\alpha\beta}$+$HIER$+LB also depends on the additional parameter α, which is responsible for the closeness of the final results to some initial predictions. Table 6.2 shows that the optimal value of $\alpha = 200$. Indeed, varying α from 1 to 200 gives a consistent increase in performance (Figure 6.10). Low values of α are too restrictive and do not allow the output values to differ much from the initial predictions. Our result is in contrast to the study of Goldberg and

Zhu (2006), which reported that optimal $\alpha = 2$. As we currently do not have an explanation for this difference, we plan to conduct experiments with the same data and in the same experimental setup used in Goldberg and Zhu (2006) as future work.

As a result of our sensitivity study, each algorithm has one or more constraints on its parameter values. This constraints should be satisfied in order to obtain good performance:

- LP_{γ}+LB has only one constraint: $\Delta_l = 0.5$.
- $LP_{\alpha\beta}$+$HIER$+LB has a similar restriction on Δ_l with the only difference being that $\Delta_l \geq 0.3$ yields accurate results. α is also an important parameter of this algorithm and its values should be close to 200.
- $RANK$+$HIER$ is very sensitive to 2 out of its 3 parameters, β and Δ_l. Similarly to the other two algorithms, the optimal value of $\Delta_l = 0.5$. A significant drop in performance observed for high values of β imposes a strong constraint on this parameter, which forces β to belong to the interval $[0.2, 0.5]$.

Interestingly, none of the algorithms revealed sensitivity to variations of k_u.

6.5 Extrinsic evaluation of similarity metrics

In Section 4.2.2, feature-based and unit-based document representations were introduced. We conducted an intrinsic evaluation of their different

components and concluded that hybrid representation, which uses document features, PWP, PSP and TitlePWP, approximates sentiment similarity best (Section 4.2.2.4). In this section, we verify extrinsically the choice of our similarity metric. The analysis is carried out for the binary and multiclass cases and we use LP_{γ}+LB as it performed best for both cases. Due to the many possible combinations of document representation components, we compare their impact by adding them one by one in the same order as in Section 4.2.2.4. The results are reported for each individual domain and are compared with the corresponding baselines and upper bounds.

6.5.1 The binary case

The binary accuracies show interesting regularities (Figure 6.11). Only the PWP component is enough to surpass the B-line for relatively small sizes of labelled data. Adding TitlePWP followed by PSP further improves the accuracy, which approaches and at times surpasses the upper bound levels for simpler domains (the PWP+TitlePWP and PWP+TitlePWP+PSP graphs in Figure 6.11). According to the intrinsic evaluation, the most accurate approximation to sentiment similarity includes PWP, PSP, TitlePWP and document features. This combination, denoted as ALL in Figure 6.11, also performs best in the extrinsic evaluation. Moreover, it surpasses the upper bound with approximately 50-100 labelled examples and continues improving when more labelled data is added. The differences between the ALL graphs and U-bounds become statistically significant for labelled data sizes greater

than 300-400 documents. However, the PWP+TitlePWP+PSP combination outperforms ALL for small sizes of labelled data (≤ 100). This could be due to the negative effect of document features. Indeed, feature-based document representations are sparse and, when little labelled data is available, the chance of finding labelled documents whose lexicons are similar to a given test document is low. Document features start playing an important role when more labelled data is added. Therefore, the accuracy graph, which exploits only the feature-based representation, has a pronounced upwards trend with low accuracy for small numbers of labelled examples. In contrast, accuracies computed for unit-based representations only are nearly parallel to the X-axis, suggesting their independence of the amount of labelled data. Unlike the feature-based representation, the unit-based representation does not depend much on the number of labelled examples, as its effectiveness is mostly determined by the accurate estimation of the sentiment strengths of the corresponding document units. It should be noted that the accuracy loss between the ALL and PWP+TitlePWP+PSP graphs is higher for BO, DV and MU, which implies the greater contribution of document features to the graph-based results for more complex domains. The BO, DV and MU domains are lexically richer and a domain-independent resource such as SO-CAL cannot cover the diversity of their lexicons.

Finally, we evaluate the contribution of titles to the graph-based performance. We consider this important for two reasons. First, this information is not always available as it depends on the text genre, and

so, it is necessary to estimate the loss of accuracy when this component is omitted. Second, it is not common to use titles to help sentiment classification; at least, we are not aware of any study which analyses the impact of titles on the final results. However, it is important to exploit all available knowledge if it helps to improve performance. Figure 6.11 shows that the difference between the ALL and PWP+PSP+Feature-based (i.e., without the TitlePWP component) graphs is significant and is equal to 3-4 ppt. This difference does not vary much across domains, suggesting that titles have a similar effect on all domains.

6.5.2 The multiclass case

For the multiclass case we report the $\bar{F}_1$ values instead of accuracies (Figure 6.12). As for the binary case, the hybrid similarity measure yields the best graph-based performance when more than 50-100 labelled examples are used. In this respect, BO demonstrates unusual behaviour because its ALL graph surpasses all other graphs only when the number of labelled examples is 300. When few labelled documents are available the PWP+TitlePWP+PSP combination usually gives more accurate results.

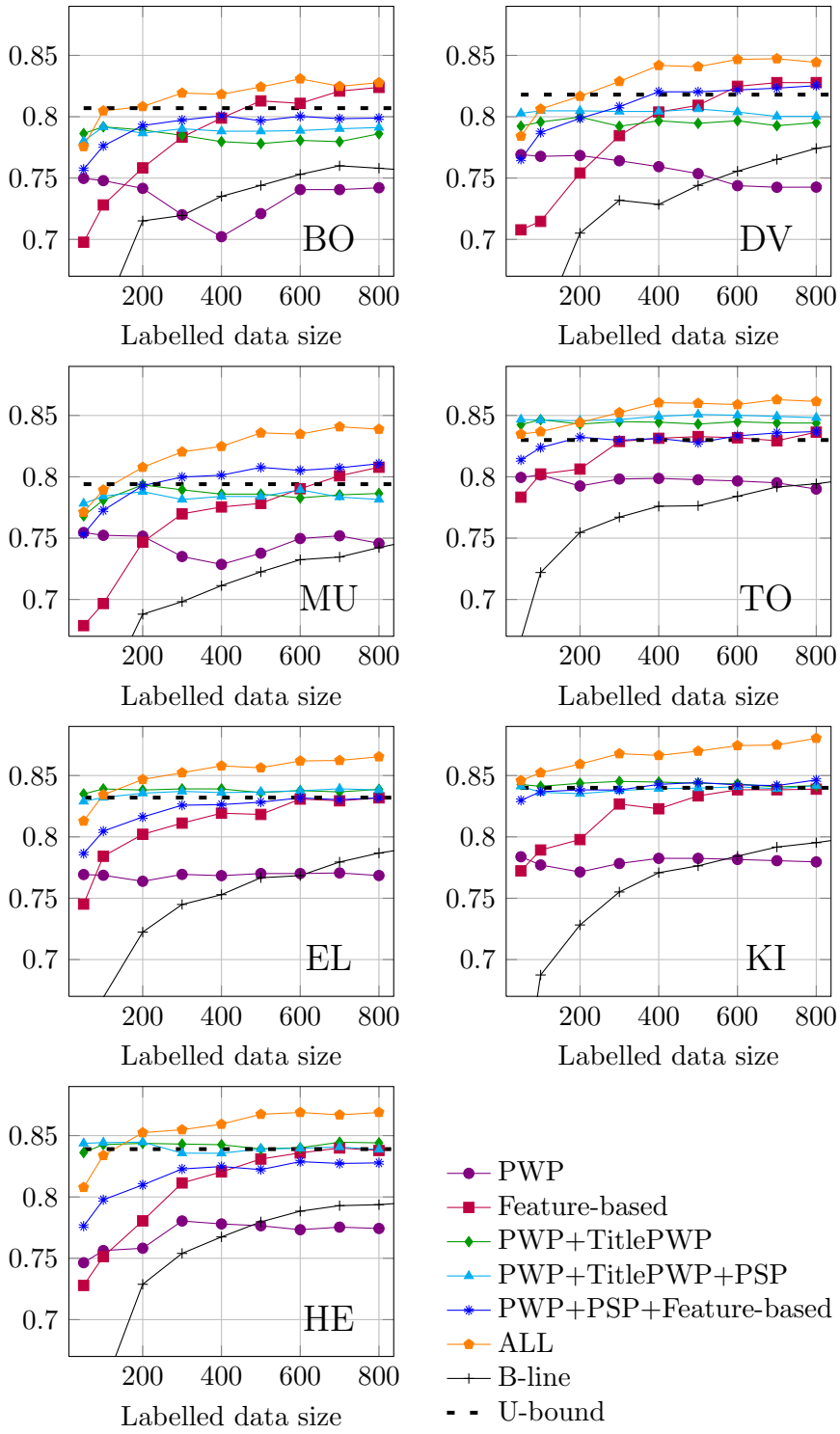

Figure 6.11: The effect of different document representation components on the accuracy of LP_{γ}+LB (binary case).

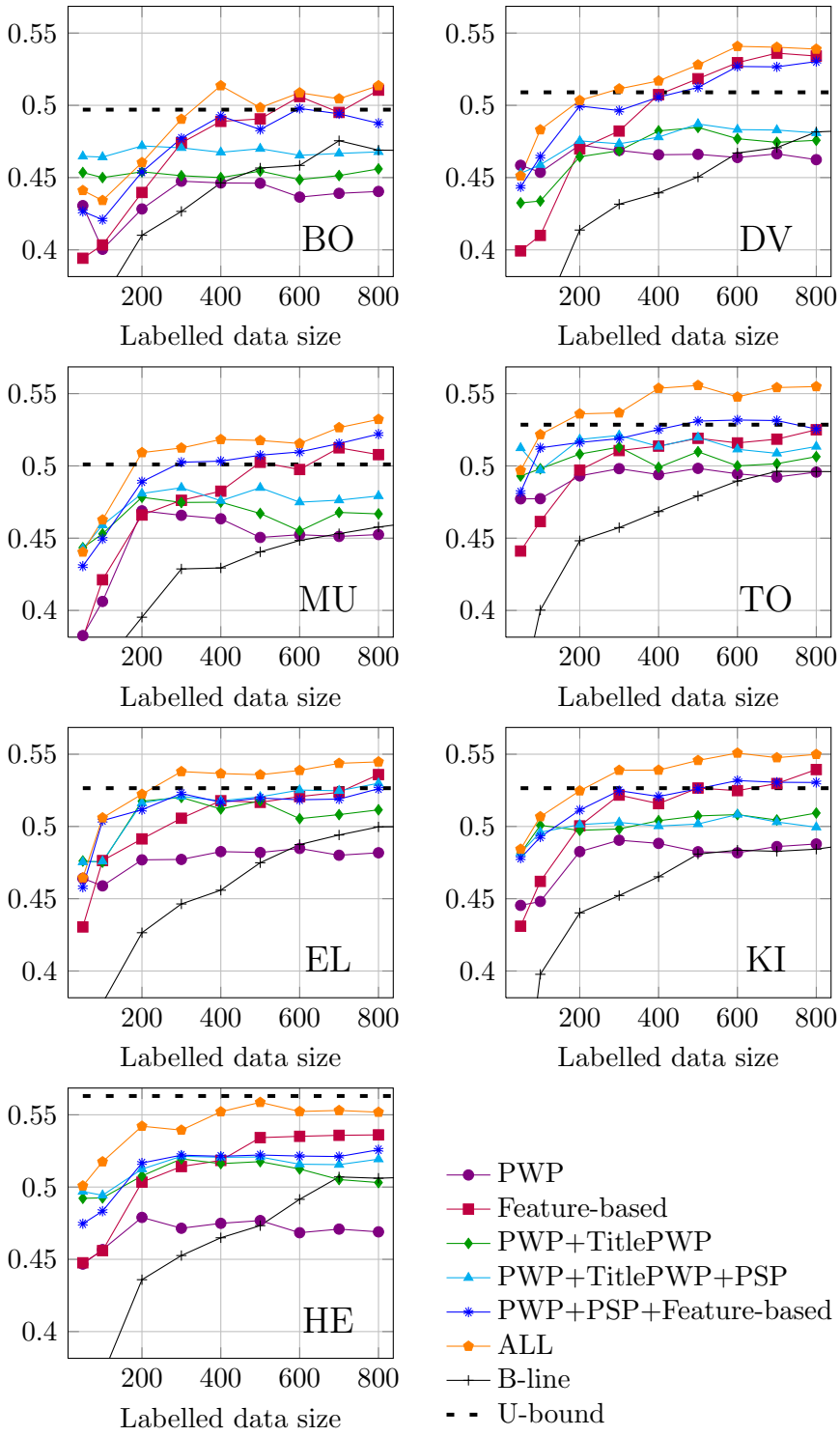

Figure 6.12: The effect of different document representation components on the $\bar{F}_1$ of LP_γ+LB (multiclass case).

Comparing Figures 6.11 and 6.12, there are a number of important contrasts between the binary and multiclass cases. First, the differences between the ALL and PWP+PSP+Feature-based graphs are smaller, which indicates a decreased effect of titles for all domains. Titles are good indicators of polarity but they are less helpful for distinguishing between finer-grained sentiments. This can be due to the title length, which restricts its expressiveness unless the opinion is very strong. Second, the drop in performance between the PWP+TitlePWP+PSP and ALL graphs is much bigger for all domains except for EL, which implies an increased impact of document features. When the number of labelled examples reaches 400, all feature-based graphs outperform the graphs where no document features are involved. Therefore, feature vectors better reflect the sentiment strengths of documents than the SO-CAL-based percentages of positive/negative words and sentences.

6.6 The effect of the adapted SO-CAL dictionaries

In Section 4.2.2.3, we described our strategy of adapting SO-CAL to the genre of product reviews. As a result, we found 65 new words which are discriminative in the sentiment sense and appear frequently in at least 3 domains out of 7. In this section, we evaluate the effect of these new sentiment markers on graph-based performance. As before, we use LP_{γ}+LB to conduct our experiments.

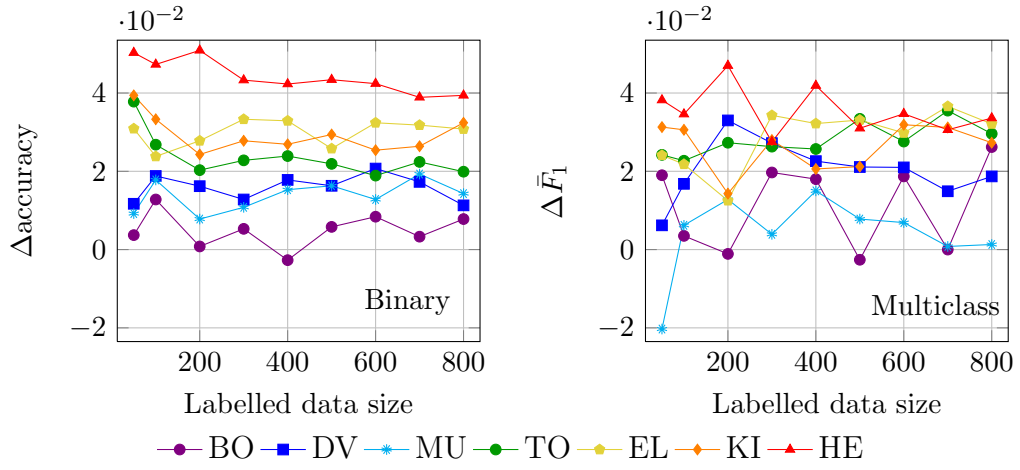

Figure 6.13: Percentage point differences between results with initial and adapted SO-CAL dictionaries.

In Figure 6.13, the differences in results for the initial and adapted SO-CAL dictionaries are displayed. Initially, these differences seem very high given that we did not expect a few sentiment markers to have such an influence on the algorithm performance. However, an analysis of their occurrence in our data revealed that the new sentiment markers have a high document frequency, indicating that at least one of these words occurs in 55-65% of documents (Table 6.3). Moreover, they constitute a substantial part of the sentiment-bearing words in the data. According to Table 6.3, the average percentage of the new sentiment markers out of all sentiment words in the documents varies from 7.5 ppt for complex domains to 16 ppt for simple domains. This is in agreement with Figure 6.13, which shows that the SO-CAL adaptation has the greatest effect on the TO, EL, KI and HE domains.

Characteristics	BO	DV	MU	TO	EL	KI	HE
Document frequency	1188	1205	1125	1199	1328	1257	1268
Average percentage, %	7.5	7.6	7.6	13.0	16.0	14.9	15.1

Table 6.3: Document frequency of the new sentiment words and their average percentage out of all sentiment words in documents.

6.7 Comparison with other semi-supervised approaches

In this section, we compare the most successful graph-based algorithm, LP_γ+LB, with several state-of-the-art approaches. Due to the popularity of the multi-domain dataset, various semi-supervised sentiment classification methods have been tested on these data. As we are not aware of any multiclass classification algorithm applied to these data, our comparison is carried out for the binary case only. We compare LP_γ+LB with the following methods:

- MECH - "mine the easy, classify the hard" (Dasgupta and Ng, 2009);
- CO-TRAIN - co-training with personal and impersonal view classifiers (Li et al., 2010a);
- ADN - active deep network (Zhou et al., 2013);
- IADN - information ADN (Zhou et al., 2013).

It should be noted that we do not reimplement these methods but use the accuracies reported in the papers and, therefore, the evaluation setup is not the same for all methods. MECH, ADN and IADN were tested with 10-fold

cross-validation, while we used 5-fold cross-validation for all our experiments. The higher number of folds implies larger training data size, which, in turn, can produce higher accuracy levels. Therefore, a direct comparison between accuracies given by the different approaches might be slightly unfair for the graph-based algorithms. Since none of the reference methods used either lexical resources or review titles, we report the LP_γ+LB accuracies for two sentiment similarity metrics: feature-based and hybrid. The corresponding algorithms are further referred to as the feature-based LP_γ+LB algorithm and the hybrid LP_γ+LB algorithm respectively. As a reminder, the feature-based metric does not require any information in addition to the review texts. In contrast, the hybrid metric also exploits the SO-CAL dictionaries and review titles, and is the measure which performed best according to our intrinsic and extrinsic evaluations (Sections 4.2.2.4 and 6.5).

MECH (Dasgupta and Ng, 2009) combines spectral clustering with active learning (see Section 2.3.3). The authors report the accuracy for 100 and 500 labelled examples manually annotated during the active learning step. According to Table 6.4, even the feature-based LP_γ+LB algorithm significantly outperforms MECH with an average difference of 6 to 8 ppt, depending on the labelled data size. The difference between MECH and the hybrid LP_γ+LB algorithm is much higher: 15 ppt for 100 labelled documents and 9 ppt for 500 labelled documents.

CO-TRAIN (Li et al., 2010a) implements the co-training approach with personal and impersonal view classifiers (see Section 2.3.3). Although

Data size	Method	BO	DV	EL	KI	average
100	MECH	0.621	0.627	0.706	0.741	0.674
	CO-TRAIN	0.626	0.495	0.700	0.786	0.652
	ADN	0.690	0.716	0.768	0.775	0.737
	IADN	0.697	0.722	0.779	0.782	0.745
	LP_γ+LB (Feature-based)	**0.729**	0.715	0.784	0.789	0.754
	LP_γ+LB (Hybrid)	**0.799**	**0.806**	**0.832**	**0.851**	**0.822**
300	CO-TRAIN	0.716	0.655	0.782	0.833	0.746
	LP_γ+LB (Feature-based)	**0.783**	**0.785**	**0.811**	0.827	**0.802**
	LP_γ+LB (Hybrid)	**0.816**	**0.832**	**0.856**	**0.856**	**0.840**
500	MECH	0.735	0.734	0.775	0.784	0.757
	LP_γ+LB (Feature-based)	**0.813**	**0.809**	**0.818**	**0.833**	**0.818**
	LP_γ+LB (Hybrid)	**0.824**	**0.845**	**0.860**	**0.859**	**0.847**

Table 6.4: Comparison of the results (accuracies) given by the feature-based and hybrid LP_γ+LB algorithms and four state-of-the-art semi-supervised approaches (the graph-based accuracies outperforming the best state-of-the-art results with a significance level of 0.05 are highlighted).

it achieves comparable accuracies to those of the feature-based LP_γ+LB algorithm for the KI domain, it yields considerably worse results for all other domains, especially DV. Moreover, the method seems to be extremely data sensitive, as the accuracy varies substantially from one dataset to another. For example, in comparison to MECH, it performs better for BO and KI, but its accuracy for DV drops so drastically that the accuracy averaged over domains is 2 ppt lower than for MECH. In general, CO-TRAIN appears to work reasonably well for simpler domains, especially when more labelled examples are involved.

ADN and IADN (Zhou et al., 2013) combine active learning with deep learning (see Section 2.3.3). IADN additionally exploits information density when choosing a review for manual annotation during the active learning

step. According to Table 6.4, ADN, IADN and the feature-based LP_γ+LB algorithm demonstrate comparable accuracies for all domains except for BO, where the graph-based approach achieves significantly better results. This means that even the knowledge-poorer version of LP_γ+LB shows a small advantage over ADNs. The hybrid LP_γ+LB algorithm clearly outperforms both ADN and IADN with quite a substantial difference of 7-10 ppt. In summary, the comparative analysis revealed not only the high effectiveness of the graph-based algorithms but also their ability to perform equally well for all datasets.

6.8 Summary

To conclude we briefly address all questions stated at the beginning of the chapter. Normalisation of the output results is beneficial for semi-supervised graph-based learning (Figures 6.3 and 6.6). LB yields the best multiclass classification results, with a substantial advantage over CMN, which gives only a rather modest gain. For binary classification, CMN and LB have similar accuracies due to the balanced class distribution of the dataset.

The $HIER$ probability combination rule consistently brings a small improvement in performance for all algorithms (Figure 6.6). For $RANK$, the $HIER$ configuration achieves the best performance in comparison with the other configurations. However, when LP_γ and $LP_{\alpha\beta}$ are considered, the advantage of $HIER$+LB over LB is disputable. The former configuration

yields higher results for larger amounts of labelled data, whereas the latter performs better when few labelled examples are available.

The binary classification results clearly indicate the advantage of LP_{γ}+LB over the other algorithms and configurations (Figure 6.4). In contrast, the multiclass experiments showed that all three variants of LP can deliver accurate results if used in the configuration which benefits them most (Figure 6.7). The most successful algorithm configurations were found to be LP_{γ}+LB, $LP_{\alpha\beta}$+$HIER$+LB and $RANK$+$HIER$. Since each LP variant has its advantages and shortcomings, the choice of the best algorithm configuration should be made on the basis of the availability of labelled data and the requirements of the given task. For example, LP_{γ}+LB gives the best results when few labelled examples are given, while $LP_{\alpha\beta}$+$HIER$+LB reaches the highest performance for large amounts of labelled data. $RANK$+$HIER$ is beneficial when there is a preference for higher accuracies over a balanced final class distribution.

Graph-based algorithms can be highly effective with a relatively small amount of labelled data. In particular, binary performance with only 100 labelled examples approaches the upper bound of accuracy (Figure 6.4). A comparison of the multiclass classification results is more difficult as we have to take into account two estimates: accuracy and $macroF_1$ (Figure 6.6). While $RANK$+$HIER$ achieves the upper bound of accuracy with 400 labelled documents, LP_{γ}+LB does not approach it even when 800 labelled examples are available. At the same time, LP_{γ}+LB surpasses the upper

bound of $macroF_1$ with only 100 labelled documents and it further increases up to the level of 0.5, which is $\approx$ 5 ppt higher than the upper bound. In contrast, $RANK{+}HIER$ gives more modest $macroF_1$ results, although it reaches the upper bounds when 400 labelled documents are used. Overall, if the mean of accuracy and $macroF_1$ is considered, both $RANK{+}HIER$ and $LP_\gamma{+}LB$ achieve the upper bound levels with 300 labelled documents (Figure 6.7).

All three graph-based algorithms are reasonably insensitive to variations of k_u and Δ_l (Figure 6.9). Although their performance drops for low values of Δ_l, this behaviour is stable for all labelled data sizes, which means that the choice of $\Delta_l = 0.5$ will always guarantee accurate results. The variations of the parameter β have a different effect on the graph-based algorithms. While $LP_\gamma{+}LB$ and $LP_{\alpha\beta}{+}HIER{+}LB$ show very little fluctuation, the performance of $RANK{+}HIER$ changes significantly with variations of β. The best β values depend on the labelled data size, which impedes us from choosing its optimal value for all data sizes.

Extrinsic evaluation of different sentiment similarity measures confirmed the high effectiveness of the hybrid similarity measure for both binary and multiclass cases (Figures 6.11 and 6.12). The similarity measure based on document units is almost insensitive to the amount of labelled data and its impact is especially valuable when labelled data is scarce. In contrast, the similarity measure based on document features is highly dependent on the number of labelled examples. Document features can degrade graph-

based performance when labelled data is insufficient, but their positive effect increases substantially with the growth of the labelled data size. We also established the significant contribution of review titles to graph-based results, especially for the binary case.

Adapting the SO-CAL dictionaries to the review genre considerably improved the overall performance of the algorithms by $\approx$ 2-3 ppt on average, depending on the domain (Figure 6.13). This procedure was most beneficial for simpler domains.

The graph-based algorithms demonstrated a substantial improvement over four state-of-the-art semi-supervised methods (Table 6.4). The average accuracy gain of the best graph-based results achieved with the hybrid similarity measure is always statistically significant and never goes below 8 ppt. Even the feature-based LP_{γ}+LB algorithm, which does not use review titles and the SO-CAL dictionaries, outperforms the state-of-the-art approaches, although the improvement is not statistically significant compared to ADN and IADN. This high performance of graph-based algorithms, and the ease with which they can be exploited, should lead to an increase in their use by the sentiment analysis research community.

It is worth pointing out another important outcome of our semi-supervised experiments. There is no significant difference between the results given by the LP variants in the basic configuration (Figure 6.1 and 6.2), which implies that the modifications they offer do not really affect performance. In contrast, sentiment similarity seems to have a substantial

impact on the results. For example, the hybrid similarity measure yields an improvement of up to 10 ppt compared to the similarity measure based on document features conventionally used for graph-based learning (Figures 6.11 and 6.12). This provides further evidence for the conviction shared by many researchers in the field that quality of modelling of the data in graph construction is more crucial than the inference algorithm used.

Chapter 7

Cross-domain experiments

This chapter presents the evaluation of the graph-based sentiment analysis system in cross-domain settings. As for semi-supervised settings, the four *LP* variants in different configurations are explored and compared. We also examine sensitivity of the graph-based algorithms to parameter variations and analyse the impact of different similarity measures on the final results. Although a similar study was done in semi-supervised settings, it is necessary to repeat the experiments in the new setup due to important differences between semi-supervised and cross-domain settings.

In the cross-domain task, labelled data from a source domain can be very different from target data. In Chapter 5, the effect of domain similarity on the cross-domain performance of two classifiers (SVMs and VP) was observed. We foresee that domain characteristics may also have a strong influence on the cross-domain graph-based results. In addition, we do not exclude the possibility that optimal parameter values and sensitivity to parameter variations may also depend on the characteristics of source-target domain pairs. In light of this, one of the main objectives of this chapter is to examine the effectiveness of cross-domain graph-based learning given certain

characteristics of source-target domain pairs. This will contribute to the third goal of the thesis: to develop guidelines which help to choose between semi-supervised and cross-domain approaches given the availability of labelled data.

Taking into consideration the above issues, our cross-domain experiments address the following questions:

1. Does cross-domain graph-based learning benefit from normalisation and the hierarchical probability combination rule?

2. Which graph-based algorithm and algorithm configuration yields the best overall performance?

3. Do the cross-domain results depend on source and target domain characteristics?

4. Given a source-target pair, are the cross-domain graph-based approaches able to achieve a performance comparable to that for in-domain classification?

5. Are the graph-based algorithms sensitive to variations of their parameters and does sensitivity depend on source and target domain characteristics?

6. To what extent is the best performing similarity metric sensitive to the source and target domain characteristics?

7. How do the graph-based algorithms perform in comparison to prominent cross-domain methods?

8. What would be the best strategy when choosing between semi-supervised and cross-domain graph-based learning?

The reminder of the chapter is organised as follows: Section 7.1 describes the cross-domain evaluation setup, and all the subsequent sections focus on the questions stated above. Section 7.2 addresses questions 1-4, Section 7.3 question 5, Section 7.4 question 6 and Section 7.5 question 7. The evaluation results are summarised in Section 7.6. Finally, Section 7.7 focuses on one of the central questions of the thesis expressed by question 8.

7.1 Experimental setup

The seven datasets give 42 combinations of source-target domain pairs and, therefore, 42 experiments. Similarly to the semi-supervised settings, the cross-domain experimental setup includes two stages: parameter tuning and algorithm testing. We randomly extract 400 examples from the target data and use them as the development dataset for tuning the parameters α, $\beta(\gamma)$, k_u and k_l. The parameter search is run over the following ranges:

- $\alpha \in \{1, 2, 5, 10, 50, 100, 200\}$,
- $\beta \in \{0.2, 0.5, 1, 2, 5\}$,
- $k_u \in \{5, 10, 20, 50, 100, 200\}$,

- $k_l \in \{10, 20, 50, 100, 200, 400\}$.

Initial values for the unlabelled documents required by $LP_{\alpha\beta}$ are obtained using the baseline classifier trained on the source data. The selection for optimal parameter values is based on two criteria: maximisation of the mean F-score $\bar{F}_1$, averaged over all source-target domain pairs, and minimisation of the $\bar{F}_1$ variance over the pairs. This reflects our preference for parameter values which produce equally good results for all source-target domain pairs, as we want to ensure that optimal values will also work well for unseen domains.

7.2 The impact of normalisation and the hierarchical probability combination rule

In this section, we compare the four LP variants and their configurations in order to establish the most successful algorithm and its configuration. The experiments are carried out separately for binary and multiclass classification.

7.2.1 The binary case

Figure 7.1 presents the accuracies averaged over source domains when the target domain is fixed. LP, LP_γ and $LP_{\alpha\beta}$ display very similar behaviour. Their CMN and LB configurations yield the same accuracy and always demonstrate a slight advantage over the basic configuration and the upper bound. Similarly to the outcomes of the semi-supervised experiments, the basic configuration is the least successful. Nevertheless, it still delivers

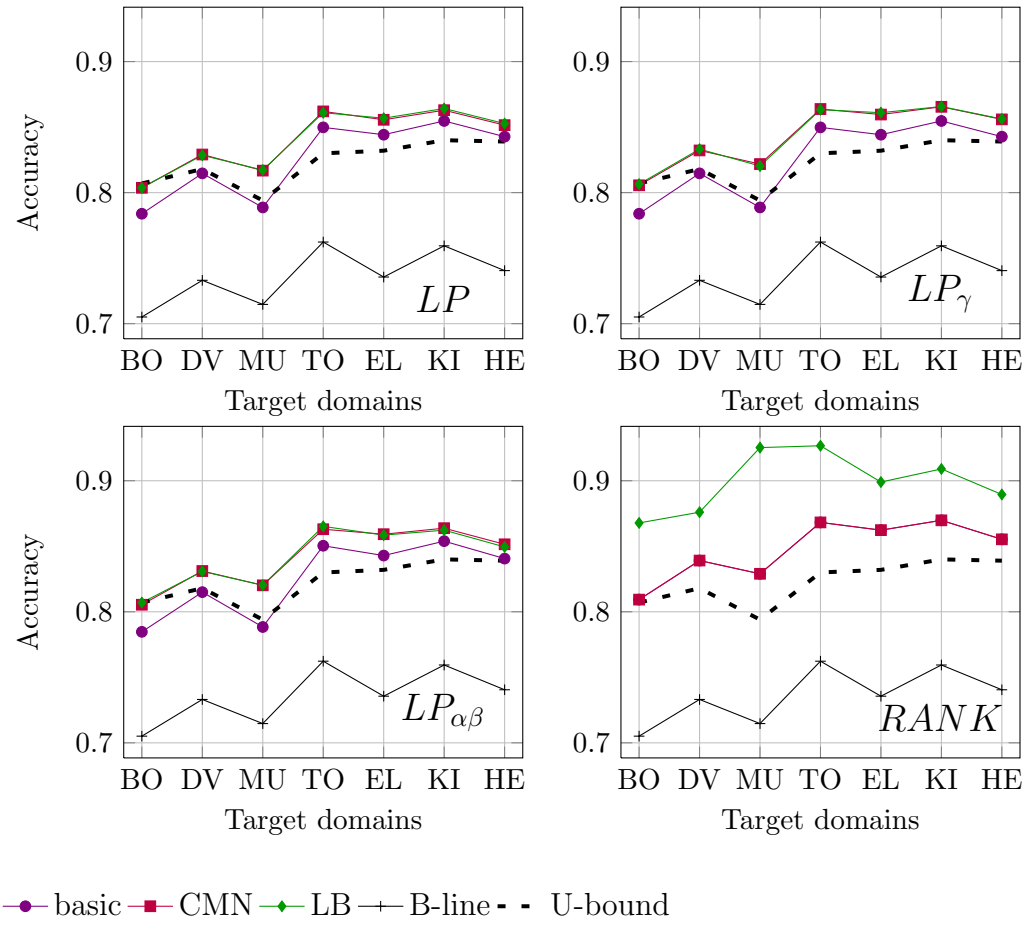

Figure 7.1: Accuracy averaged over source domains with target domains fixed, for different algorithms and their configurations (binary case).

reasonable results, surpassing the upper bound for simpler target domains. In contrast to the first three algorithms, $RANK$ behaves very differently. Although its performance in the basic configuration is still comparable to the LP_γ+CMN results, its LB configuration gives outstanding results, improving the upper bound by 5-10 ppt depending on the target domain. However, we cannot offer an explanation for this performance of $RANK$+LB other than that it may be due to the data characteristics and, therefore,

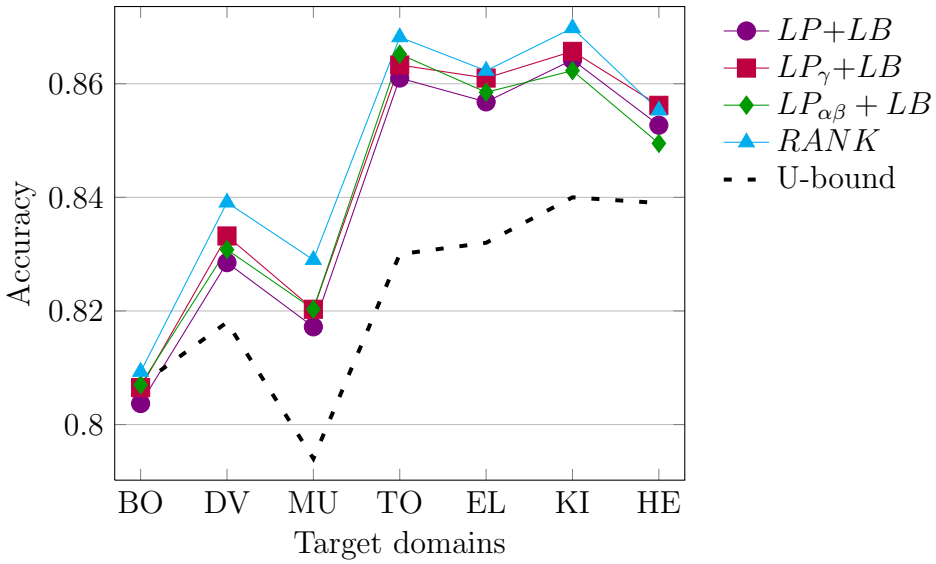

Figure 7.2: Accuracy averaged over source domains with target domains fixed, for the most successful algorithms and normalisation techniques (binary case).

cannot be generalised for all datasets. $RANK$+LB performed quite poorly in semi-supervised settings, and, as discussed below, it does not stand out among other configurations for the multiclass case either. Therefore, taking into account its excellent performance for semi-supervised settings, we consider $RANK$ to be the best configuration.

The most accurate configurations of each LP variant are displayed in Figure 7.2. $RANK$ consistently outperforms the other algorithms, although the difference is not significant. In Figure 7.3, the accuracy and MSE of the two best algorithms, $RANK$ and LP_γ+LB are shown[1]. As expected,

[1] There is, in fact, no substantial difference between LP_γ+LB, LP+LB and $LP_{\alpha\beta}$, but we preferred the first method as it achieved the highest performance in semi-supervised settings.

similarity between source and target datasets always ensures excellent results. However, domain complexity influences the maximum accuracy that each dataset can achieve and, therefore, complex target domains usually perform worse than simple domains. When source and target data are dissimilar, domain complexity becomes more crucial. Indeed, for a simple target domain the results do not vary much from one source domain to another. Even when a source domain is very different from a target domain, the accuracies are only ≈2 ppt lower than those given by similar source domains. This indicates that cross-domain graph-based learning can be very successful for simple target data independently of the source data characteristics. Unlike simple target domains, complex target domains trained on dissimilar data give a substantial drop in accuracy. Therefore, for complex target domains, the cross-domain results are determined by both source and target datasets. Overall, the best accuracies for simple target domains are 2-3 ppt higher than those for complex domains. The MSE values are in agreement with the accuracies and support the differences indicated between simple and complex domains. In particular, the MSE values for simple target domains are lower and have smaller variations from one source domain to another.

Figure 7.3 does not give an understanding of how good performance on individual domains is in comparison with the accuracy upper bounds. In order to present the cross-domain results and the accuracy upper bounds for all target domains in the same graph, we swap the source and target domains in Figure 7.3 and draw graphs corresponding to source domains

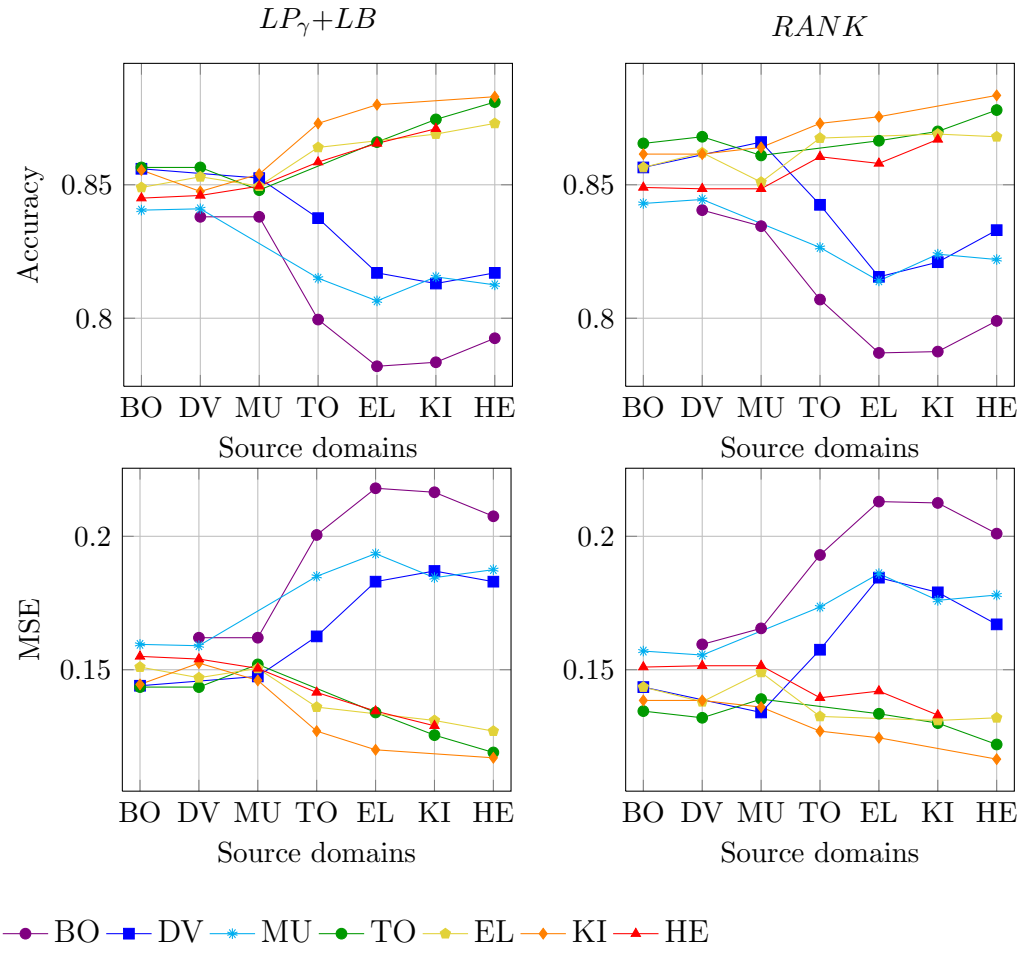

Figure 7.3: Accuracy and MSE obtained with LP_γ+LB and $RANK$ for each source-target domain pair. X-axes contain source domains, graphs correspond to target domains (binary case).

with target domains on the X-axis (Figure 7.4). This different representation allows a deeper understanding of the results. The accuracies on almost all source-target domain pairs are above the upper bound, which proves that the graph-based algorithms are highly effective for cross-domain sentiment classification. A few results that are slightly inferior to the upper bound correspond to simple source domains and the target domains of BO and

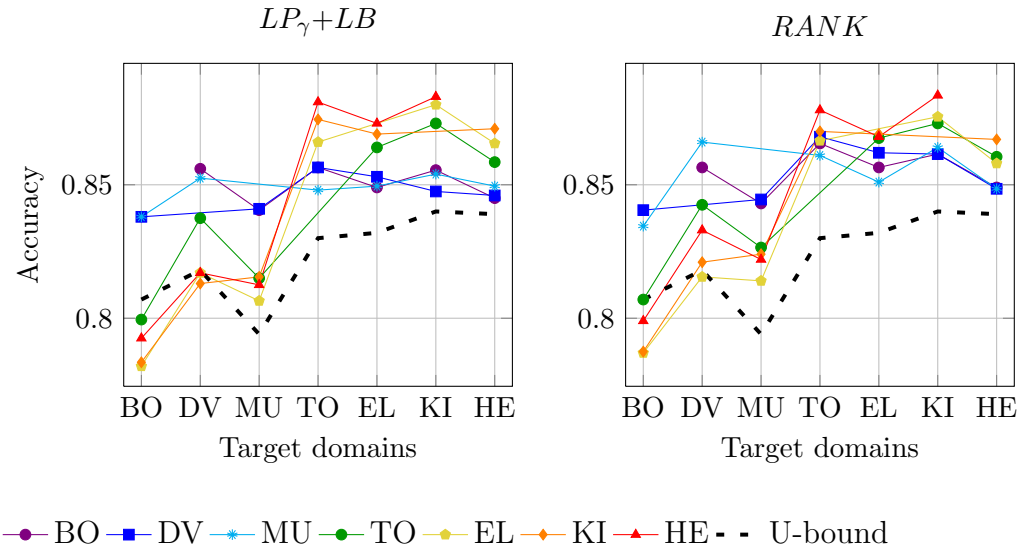

Figure 7.4: Accuracy obtained with LP_γ+LB and $RANK$ for each source-target domain pair. X-axes contain target domains, graphs correspond to source domains (binary case).

DV. This again confirms that simple source and complex target domains, when they are both dissimilar, are challenging for cross-domain graph-based learning.

7.2.2 The multiclass case

Figure 7.5 shows the performance of the four LP variants and their configurations for the multiclass case. The comparison is carried out using $\bar{F}_1$ values to establish which method is the most successful on average. Overall, the algorithms perform similarly to the semi-supervised results. The LB and $HIER$+LB configurations are the most accurate and even reach the upper bound for some target domains. However, unlike in semi-supervised settings, $HIER$+LB yields moderately better results than LB. Looking

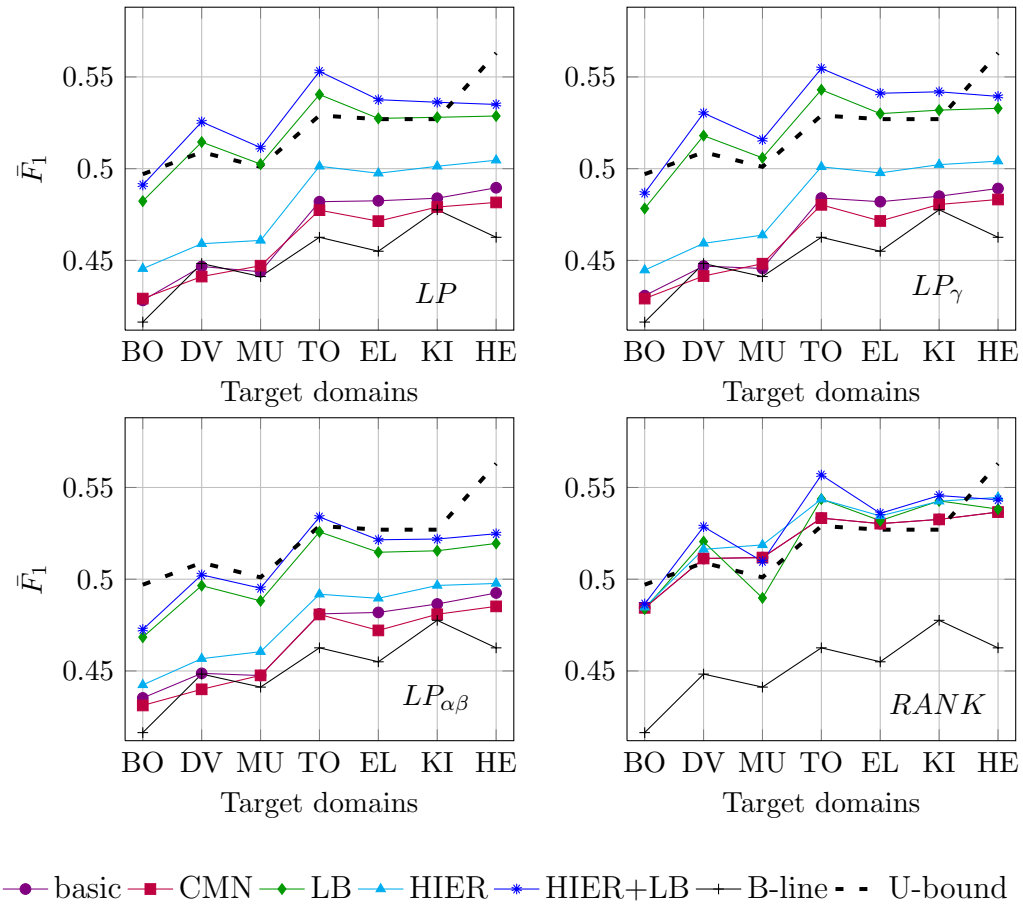

Figure 7.5: $\bar{F}_1$ averaged over source domains with target domains fixed, for different algorithms and their configurations (multiclass case).

more carefully at the semi-supervised results, we observe that $HIER{+}LB$ also outperformed LB when there was a substantial amount of labelled data. In contrast, for small amounts of labelled data the $HIER{+}LB$ configuration does not give reliable results, showing a significant decrease in performance. This was the reason we preferred the LB configuration in semi-supervised settings. Interestingly, $LP_{\alpha\beta}$ performs worst in cross-domain settings, indicating that the initial predictions add noise rather than

useful knowledge. As before, the behaviour of $RANK$ is different to other algorithms as all its configurations give comparable results with a slight advantage for $RANK{+}HIER{+}LB$ over the others.

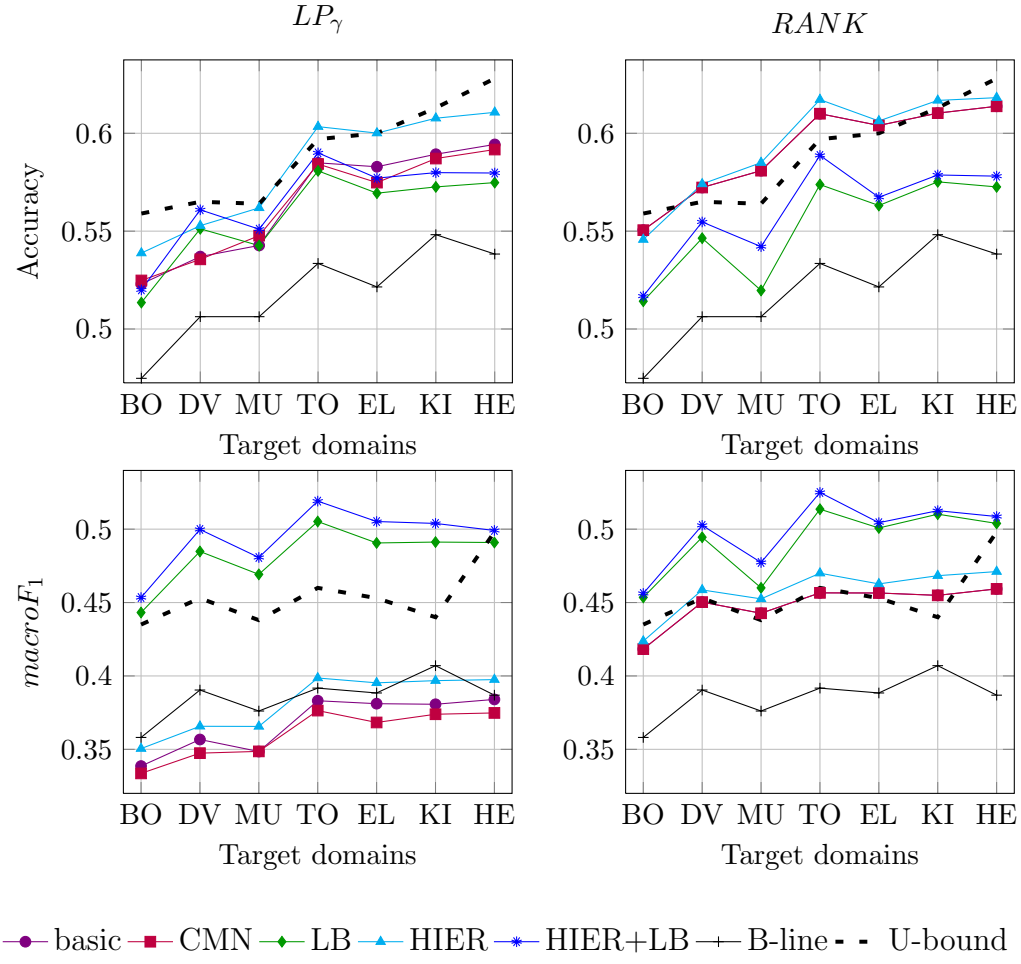

Figure 7.6: Accuracy and $macroF_1$ averaged over source domains with target domains fixed for different configurations of LP_γ and $RANK$ (multiclass case).

In Figure 7.6, we further compare the algorithms by their accuracy and $macroF_1$. Since LP and $LP_{\alpha\beta}$ give marginally lower $\bar{F}_1$ values than

LP_γ, we only display the results of LP_γ and $RANK$. In general, the algorithm configurations show the same regularities that we observed in semi-supervised settings. $macroF_1$ values are usually highest for the LB and $HIER{+}LB$ configurations and lowest for the others. In contrast, accuracies tend to be higher for the basic, CMN and $HIER$ configurations, which is especially obvious in the case of $RANK$. LP_γ yields similar accuracies in all configurations, with a slight tendency of $HIER$ for higher performance. If we use only the $\bar{F}_1$ maximisation criterion for choosing the most accurate algorithm, we would select either $LP_\gamma{+}HIER{+}LB$ or $RANK{+}HIER{+}LB$. However, they both have the disadvantage of rather moderate accuracy. $RANK{+}HIER$ would be more beneficial if accuracy is of a higher priority than $macroF_1$. Indeed, $RANK{+}HIER$ does not only surpass the accuracy upper bound but also, unlike the corresponding configuration of LP_γ, reaches the upper bound of $macroF_1$. According to Figure 7.5, $RANK{+}HIER$ is only marginally inferior to $RANK{+}HIER{+}LB$. In summary, we can distinguish two algorithms that give the most accurate results: $LP_\gamma{+}HIER{+}LB$ and $RANK{+}HIER$[2]. Since both of them have their strengths and shortcomings, the requirements of the task at hand should determine which should be preferred. For example, companies might be interested in mining moderately positive reviews to find out what minor problems users experience with their products. This means that they may want to detect as many 4* reviews as possible and if 4* reviews

[2] $RANK{+}HIER{+}LB$ is omitted due to its similar behaviour to $LP_\gamma{+}HIER{+}LB$.

are rare in the data, LP_γ+LB will be preferable. In contrast, if companies are more interested in estimating the overall satisfaction/dissatisfaction of users with their products, then $RANK$+$HIER$ will be more beneficial.

Figure 7.7 presents the accuracy, $macroF_1$ and MSE given by the two best algorithms for individual source-target domain pairs. As with the binary results, the accuracy and MSE graphs are different for simple and complex target domains. The accuracy and MSE values of simple target domains are almost insensitive to the characteristics of labelled data as they show a relatively low variation with respect to source domains. The $macroF_1$ graph of LP_γ+$HIER$+LB shows the same regularity as the accuracy and MSE graphs, although the difference in $macroF_1$ values between simple and complex target domains is less pronounced. In contrast, the $macroF_1$ values of $RANK$+$HIER$ are higher for complex source domains, and this is consistent for all target domains. This implies that if the labelled data is simple, $RANK$+$HIER$ will not learn to distinguish between different sentiment strengths and will tend to identify correctly only most numerous sentiment classes. Interestingly, the $macroF_1$ values for simple domains are higher when learning on different but more complex labelled data. In general, the $macroF_1$ values given by $RANK$+$HIER$ are much lower than those of LP_γ+$HIER$+LB. The lowest $macroF_1$ values are observed for complex target domains when the labelled data is different and simple.

Similarly to the binary case, we swap source and target domains in order to present the cross-domain performances and the upper bounds

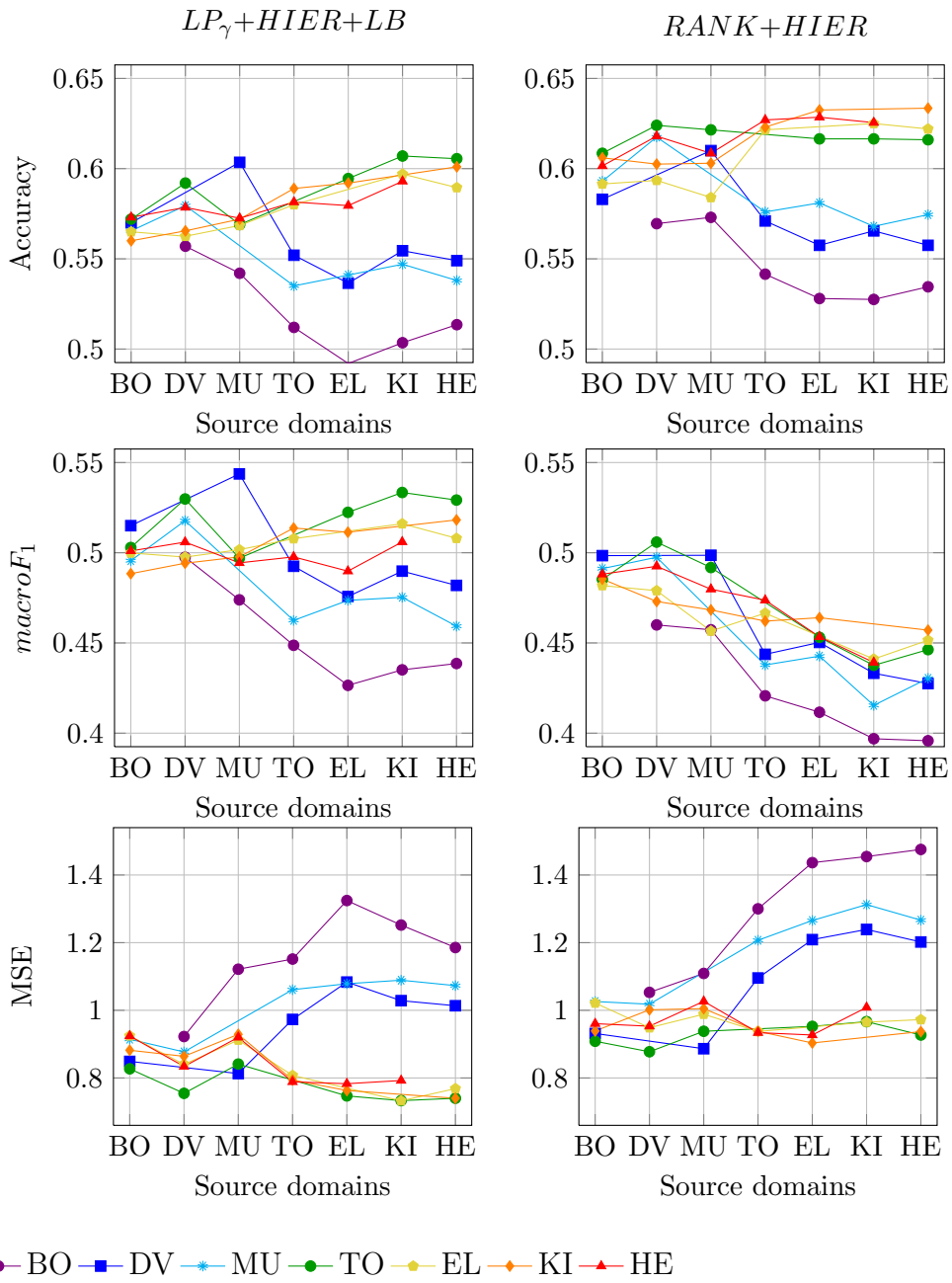

Figure 7.7: Accuracy, $macroF_1$ and MSE obtained with LP_γ+$HIER$+LB and $RANK$+$HIER$ for each source-target domain pair. X-axes contain source domains, graphs correspond to target domains (multiclass case).

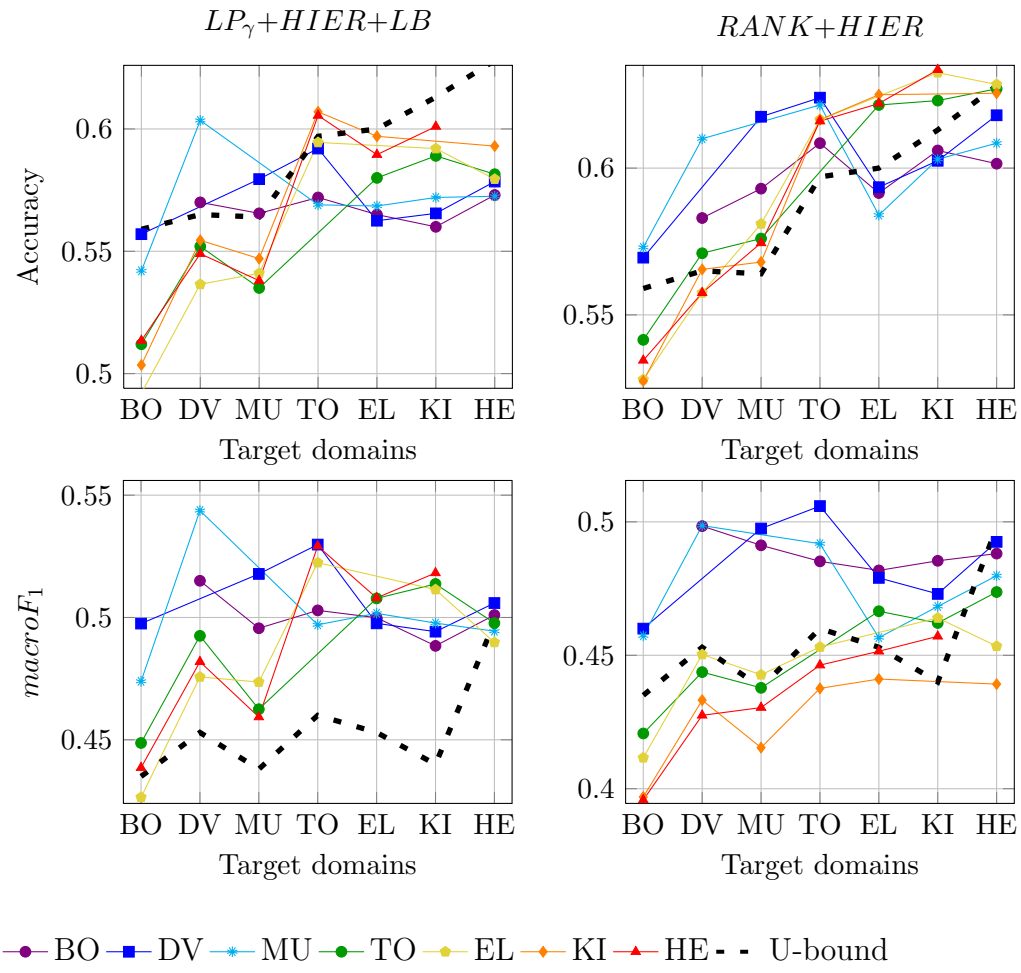

Figure 7.8: Accuracy, $macroF_1$ and MSE obtained with LP_γ+HIER+LB and RANK+HIER for each source-target domain pair. X-axes contain target domains, graphs correspond to source domains (multiclass case).

for all target domains in the same plot (Figure 7.8). The result of this merge is less clear and more difficult to interpret than for the binary case. LP_γ+HIER+LB yields lower accuracies than the upper bound for almost all domain pairs. However, its $macroF_1$ values are significantly better than the upper bound level, which coincides with our expectations based on Figure 7.6. RANK+HIER surpasses the accuracy upper bound only for similar source-

target domain pairs. For dissimilar domains, the accuracy is still close to the upper bound for all domains except for the BO target domain, which, as in the binary case, gives the worst accuracy. Concerning the $macroF_1$ values for $RANK{+}HIER$, we observe that training on complex source domains ensures $macroF_1$ values superior to the upper bound. For simple labelled data, $RANK{+}HIER$ has a tendency towards low $macroF_1$ values, which are either below or only marginally above the upper bound. Therefore, if a balanced representation of sentiment classes in the final results is important, it is more appropriate to use either $LP_\gamma{+}HIER{+}LB$ or $RANK{+}HIER$ together with complex labelled data.

7.3 Sensitivity to parameter variations

In this section, we examine the sensitivity of the two best graph-based algorithms, $LP_\gamma{+}HIER{+}LB$ and $RANK{+}HIER$, to variations of their parameters. The analysis is carried out for multiclass classification only as similar results are expected independently of the number of classes. Table 7.1 lists the optimal parameter values of $LP_\gamma{+}HIER{+}LB$ and $RANK{+}HIER$. Comparing Tables 7.1 and 6.2, where optimal values for semi-supervised settings are given, we observe that the optimal values for $RANK{+}HIER$ are almost identical in both settings. In contrast, $LP_\gamma{+}HIER{+}LB$ shows considerable variations in parameter values. For example, $\beta = 0.2$ and $k_u = 5$ in semi-supervised settings are changed to $\beta = 5$ and $k_u = 50$ in cross-domain settings. Further analysis revealed that $LP_\gamma{+}HIER{+}LB$ is

extremely sensitive to the amount of labelled data (Figure 6.9) and, thus, its optimal parameters, when chosen for all labelled data sizes, do not produce good overall results. For small amounts of labelled data, lower values of β and k_u are more beneficial. When the number of labelled examples increases, higher values of β and k_u perform better, which is in line with the results of the cross-domain experiment.

	Parameters		
Method	β	k_u	k_l
LP_γ+$HIER$+LB	5	50	200
$RANK$ + $HIER$	0.2	50	200

Table 7.1: Optimal parameter values for LP_γ+$HIER$+LB and $RANK$+$HIER$.

The evaluation of the graph-based algorithms in cross-domain settings showed that their performance depends on the similarity and complexity of source-target domain pairs. We suspect that the optimal parameter values, as well as the algorithm stability, may also be affected by characteristics of source and target domains. Therefore, the sensitivity analysis is conducted separately for four categories of domain pairs: similar pairs with simple source and target domains (SMPL-SMPL), similar pairs with complex source and target domains (CMPX-CMPX), different pairs with simple source and complex target domains (SMPL-CMPX) and different pairs with complex source and simple target domains (CMPX-SMPL). Figure 7.9 displays the sensitivity graphs for each of the three parameters: β, k_u and k_l. The graphs are built by varying one of the parameters and

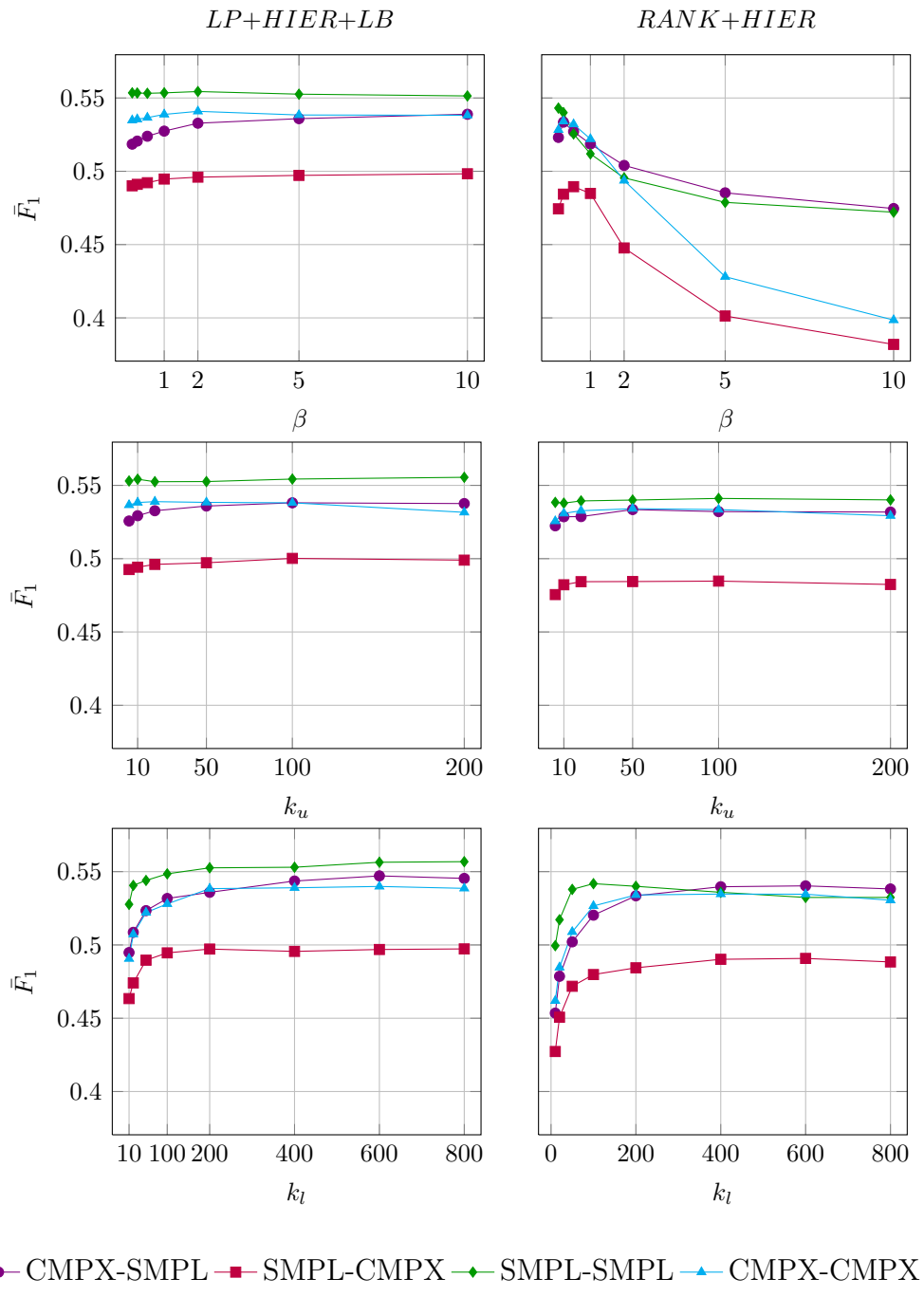

Figure 7.9: Sensitivity of LP_γ+$HIER$+LB, and $RANK$+$HIER$ to variations of their parameters. The results are presented for groups of domains, compiled by their complexity.

leaving the others fixed at their optimal values. We observe that the sensitivity of both algorithms is consistent with the semi-supervised study: the effectiveness of LP_γ+$HIER$+LB is mainly determined by the parameter k_l, while $RANK$+$HIER$ results hinge upon two parameters, β and k_l. Both algorithms perform worse when the values of k_l are low. However, $k_l = 100$ already ensures stable results approaching the algorithms' maximum performance. $RANK$+$HIER$ is highly dependent on the values of β, achieving its best results when $0.2 \leq \beta \leq 0.5$. Since this behaviour is systematic for all domain pairs, it cannot be considered as a disadvantage of the algorithm. Moreover, this optimal value is confirmed by our semi-supervised study as well as by the work of Wu et al. (2009).

There is little evidence that sensitivity is affected by characteristics of source-target domain pairs other than that the drop in performance shown by $RANK$+$HIER$ for high values of β is steeper for complex target domains. All other sensitivity graphs display similar behaviour for all source-target domain pairs and differ only in performance levels. This lack of dependence of optimal parameters on domain characteristics is a strength of cross-domain graph-based learning as it suggests that the established optimal values will also work well for unseen data.

7.4 Analysis of the similarity measures

Extrinsic evaluation of similarity measures in semi-supervised settings indicated a significant advantage of the hybrid metric, which makes use of

both feature-based and unit-based document representations (Chapter 6). It was also shown that the PWP, PSP and TitlePWP components make a considerable contribution to the semi-supervised graph-based performance. The feature-based component was also found to be important, but only for increased amounts of labelled data when feature sparseness stops being a problem. In cross-domain settings we anticipate that some document representation components might also be ineffective for certain source-target domain pairs. For example, a feature mismatch between a pair of dissimilar domains can make the feature-based component less useful. For that reason, in this section we analyse the effect of the document representation components on the cross-domain results. In particular, we aim to establish whether there is any correlation between similarity of source-target domain pairs and the similarity metric which produces the best results for this domain pair. The impact of each document representation component is evaluated in the same order as in semi-supervised settings. The experiments are conducted for the more general case of multiclass classification.

Figure 7.10 presents the $\bar{F}_1$ values given by LP_γ+$HIER$+LB for different combinations of the document representation components. As expected, the success of the feature-based similarity measure depends on the similarity of source and target domains. The cross-domain performance is substantially higher for similar source-target domain pairs and it loses about 5 ppt on average when domains are dissimilar. In contrast, when only unit-based components are involved, no significant correlation between the final

results and source domain characteristics is observed. This is not very surprising because the unit-based document representation does not contain much lexical information about documents except for their percentages of positive/negative words/sentences. Therefore, some fluctuations in performance over domain pairs could be due to discrepancies in the PSP and PWP estimations for distinct domains. When the feature-based and unit-based components are combined, the negative effect of the feature-based component for dissimilar domains is smoothed, which is especially efficient for simple target domains. This means that the insensitivity of graph-based results to source and target domain characteristics observed in previous sections is due to the positive impact of the unit-based component. Overall, in agreement with the semi-supervised experiments, the hybrid similarity measure is proved to be the most effective for all domain pairs. The contribution of document features is usually greater than that of titles as the performance drops more drastically when the former component is omitted.

7.5 Comparison with other cross-domain approaches

The multi-domain dataset has been extensively used for testing a vast number of cross-domain sentiment classification methods. For our comparison study, we select only the most prominent cross-domain techniques which have been used as a reference by many researchers in the field: structural

correspondence learning (SCL) (Blitzer et al., 2007), spectral feature alignment (SFA) (Pan et al., 2010) and the sentiment sensitive thesaurus-based method (SST) (Bollegala et al., 2011). It should be noted that these methods were designed to tackle the binary classification problem, therefore, we compare them against $LP_{\gamma}{+}LB$ and $RANK$, which performed best in our binary experiments. As in the reference studies, the results are tested on four domains: BO, DV, EL and KI.

All three reference methods aim to find representations of source features in the target domain based on co-occurrence analysis. SCL finds correspondences between domain-specific source and target features through linear modelling of their correlations with domain-independent sentiment markers which frequently occur in both domains (so-called pivot features). SFA exploits spectral clustering to align domain-specific and domain-independent features into a set of feature-clusters. SST matches source and target features through an automatically constructed sentiment-sensitive thesaurus where each lexical entry is connected to a list of related entries of the same polarity. In contrast to SCL and SFA, SST exploits multiple domains instead of just one source domain to build the thesaurus, which gives it an advantage over the other methods.

Figure 7.11 displays the results of our comparison. As SST exploits multiple sources, its performance is given only once for each target domain.

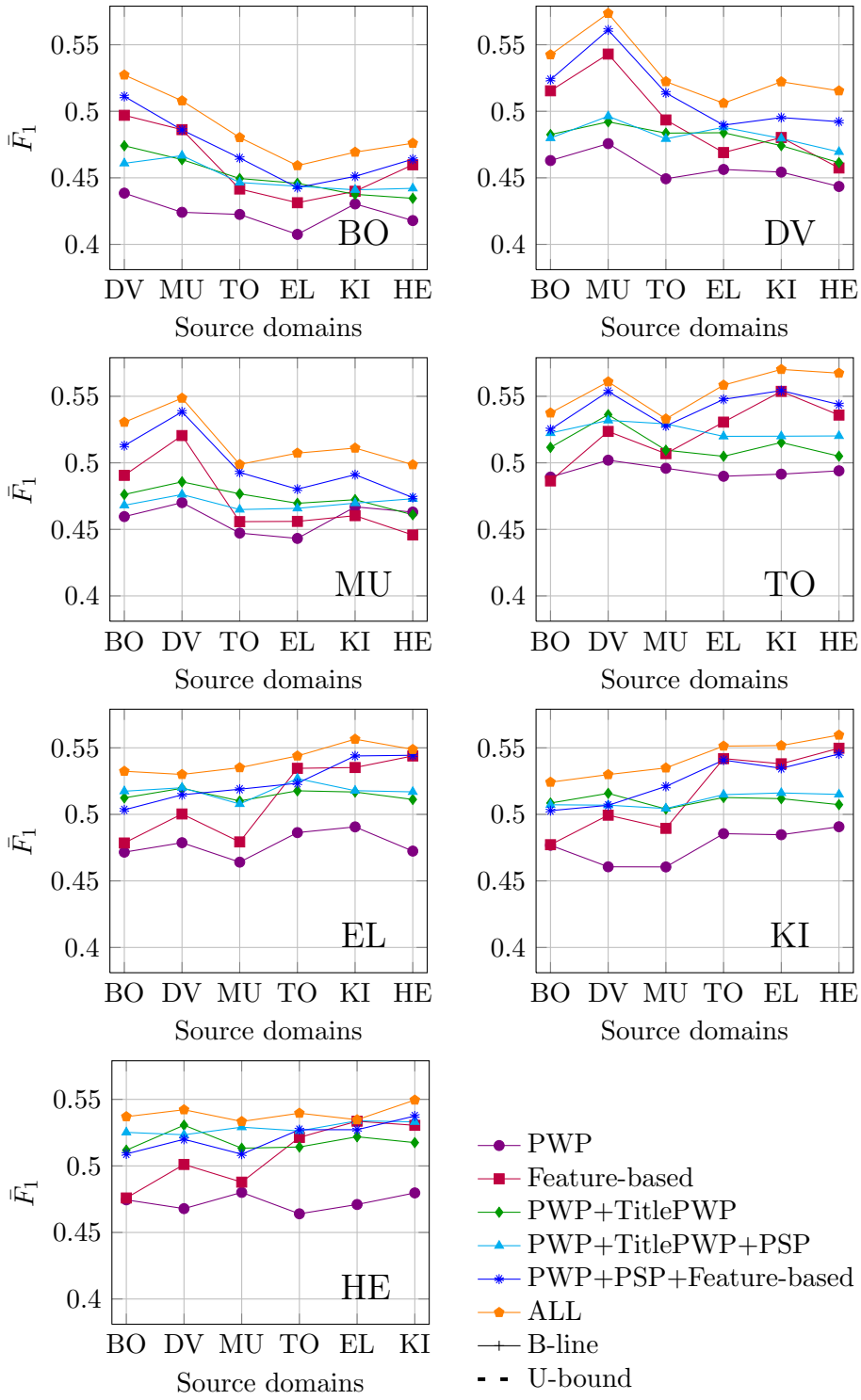

Figure 7.10: The effect of different document representation components on the cross-domain results (multiclass case).

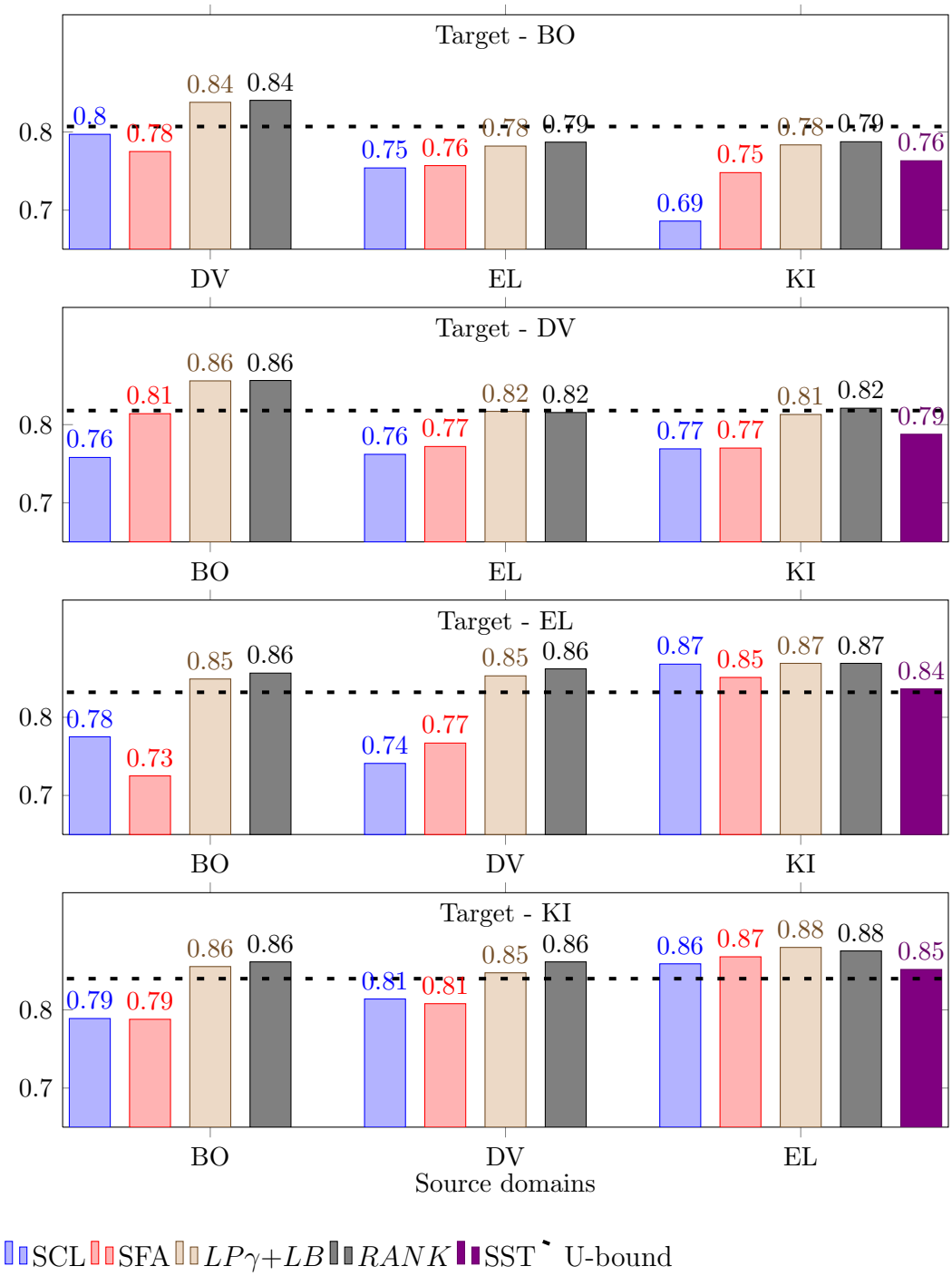

Figure 7.11: Comparison of the two best graph-based algorithms, LP_γ+LB and $RANK$, with state-of-the-art methods (accuracies).

Both LP_γ+LB and $RANK$ significantly outperform the reference methods for all source-target domain pairs. Moreover, the graph-based results on individual domain pairs are consistently better than the SST results obtained using three source domains. As mentioned previously, LP_γ+LB and $RANK$ reach or surpass the upper bound for all domain pairs except for EL-BO and KI-BO.

Besides its superior performance, graph-based learning has other advantages over the reference methods. First, SCL, SFA and SST require a substantial amount of labelled and unlabelled data to find correspondences between source and target features. This can be a shortcoming as labelled data is often limited or expensive to acquire. In contrast, a large amount of labelled data is not crucial for graph-based algorithms to perform well. Indeed, as shown by our semi-supervised experiments, the benefit of graph-based learning from additional labelled data reduces significantly when they reach approximately 500-600 examples (Figure 6.5). Second, SCL, SFA and SST are quite sensitive to source and target domain characteristics as their performance decreases considerably for dissimilar source and target domains. However, our graph-based algorithms, due to the accurate sentiment similarity measure, yield equally good results for distinct domains when the target data is simple. Even for complex target datasets, the performance is inferior to the upper bound by less than 3 ppt. Finally, SCL and SFA can only be applied for the binary classification problem and the possibility of extending them to handle multiclass classification is questionable. In

contrast, as demonstrated in this thesis, graph-based algorithms can easily be adapted to a larger number of classes[3].

7.6 Summary

To summarise the findings of our cross-domain graph-based experiments, we briefly address the issues raised at the beginning of the chapter. Normalisation and the hierarchical probability combination rule are beneficial for most graph-based algorithms. For binary classification, CMN and LB increase the accuracy by 1-2 ppt (Figure 7.1). For multiclass classification, LB brings an even higher overall gain in performance for LP, LP_γ and $LP_{\alpha\beta}$, but it only affects the $macroF_1$ values (Figures 7.5 and 7.6). The hierarchical probability combination rule consistently improves the results given by the basic configuration. Moreover, together with LB it usually delivers the best performance.

The binary classification results show a marginal advantage of $RANK$ over LP_γ+LB and $LP_{\alpha\beta}$+LB (Figure 7.2). In the multiclass experiments two best algorithms, LP_γ+$HIER$+LB and $RANK$+$HIER$, are identified. On one hand, these yield the highest $\bar{F}_1$ values, and on the other, they give contrasting performances in terms of accuracy and $macroF_1$ (Figure 7.6). LP_γ+$HIER$+LB achieves very high $macroF_1$ levels but gives a moderate accuracy. $RANK$+$HIER$, in contrast, reaches the accuracy upper bound but yields lower $macroF_1$ values in comparison with LP_γ+$HIER$+LB. As

[3] It is worth pointing out that SST could also straightfowardly be applied to multiclass classification since all the basic technique does is augment the feature representation.

pointed out in Section 7.2, the requirements of the task at hand should dictate which of these methods should be used.

We established that the cross-domain results depend greatly on the source and target domain characteristics. Domain similarity is found to be crucial as the graph-based algorithms are most effective for similar domains (Figures 7.3 and 7.7). Domain complexity also has a significant impact on the outcomes of the cross-domain task. For simple target domains, the graph-based performance does not vary substantially from one source domain to another, which indicates that it is practically independent of the similarity between source and target data. In contrast, the graph-based results for complex target domains drop drastically when source and target data are different.

For binary classification, cross-domain graph-based learning yields very high accuracies, which are superior to the in-domain results for almost all domain pairs (Figure 7.4). The outcomes of multiclass classification are difficult to interpret if we take into account both the accuracies and $macroF_1$ values. Figure 7.8 suggests that different algorithms have their advantages and disadvantages depending on which evaluation metric is considered to be more important. However, if the trade-off between accuracy and $macroF_1$ is required, the best algorithms clearly surpass the in-domain results for all target domains except BO and HE (Figure 7.5).

The sensitivity study proved the relative stability of the graph-based algorithms in cross-domain settings (Figure 7.9). As for semi-supervised

learning, $RANK$ is highly sensitive to variations of β, but the optimal value of $\beta = 0.2$ is shown to be independent of domain characteristics and coincides with the optimal value obtained in semi-supervised settings. We also established that the graph-based algorithms usually yield poor results for low values of k_l. However, $k_l \approx 200$ ensures a high performance close to the maximum effectiveness of the algorithms. Similarly to semi-supervised settings, the graph-based algorithms in cross-domain settings revealed little sensitivity to variations of k_u. Interestingly, no correlation between the optimal parameters and domain characteristics is observed. In contrast, the shapes of the $\bar{F}_1$ graphs are almost identical for all domain pairs.

The analysis of the similarity measures revealed that the feature-based document representation makes the graph-based results highly sensitive to the similarity between source and target domains (Figure 7.10). Thus, domains with a low feature overlap give much poorer performance compared to similar domains when only feature-based representation is used. In contrast, pure unit-based representations make the graph-based results independent of source and target domain characteristics. The hybrid similarity measure possesses traits of both representations. Due to the positive effect of the unit-based component, the graph-based algorithms become insensitive to the similarity between source and target data for simple target domains.

The graph-based results demonstrated an evident superiority over prominent state-of-the-art approaches (Figure 7.11). This outcome is

even more valuable when we take into account additional advantages of graph-based learning, such as the easy extension to multiclass classification and the ability to perform well with relatively small amounts of labelled data.

7.7 Semi-supervised vs. cross-domain graph-based learning

The experimental results provided by this thesis proved that graph-based learning can be highly effective for tackling the problems of semi-supervised and cross-domain sentiment classification. In particular, LP_{γ}+LB coupled with the hybrid similarity measure significantly outperformed state-of-the-art semi-supervised and cross-domain approaches in the binary classification task. But which learning setup - semi-supervised or cross-domain - should be preferred given the source data available? On one hand, cross-domain approaches have an advantage over semi-supervised approaches as they re-use already annotated data, which means that no manual effort is required. On the other hand, the cross-domain graph-based results largely depend on the characteristics of the source-target domain pairs (see Figures 7.4 and 7.8). Therefore, for dissimilar source and target domains, manual annotation of some target data and use of semi-supervised graph-based learning could help to achieve more accurate results. For example, according to Figures 6.4 and 6.7, 200-300 examples are enough to match the performance of in-domain classification in many cases.

In this section, we propose recommendations which can help to select the most pertinent learning approach given the data available. As a semi-supervised method, LP_γ+LB is used for both binary and multiclass cases. As cross-domain methods, LP_γ+LB and LP_γ+$HIER$+LB are considered for the binary and multiclass cases respectively. For all LP modifications, the hybrid similarity measure which performed best in both intrinsic and extrinsic evaluation is deployed.

Learning approaches are assessed and compared using the following two criteria:

- Maximising the accuracy (or $\bar{F}_1$)[4];
- Minimising the manual effort needed for annotation.

The decision-making process is based on two data characteristics, domain complexity and domain similarity, as they were found to have a significant impact on both semi-supervised and cross-domain performance. For ease of comparison between approaches, we illustrate the results in the same plot (Figures 7.12 and 7.13). Each plot has two X-axes. The bottom axis corresponds to the amount of labelled data and is used for the semi-supervised results. The top axis lists source domains and is used for the cross-domain results. Figure 7.12 shows that in the case of binary classification, semi-supervised learning for simple domains needs around 100 labelled examples to ensure accurate performance.

[4]To simplify the analysis, we do not prefer one result over another if they both reach the upper bound.

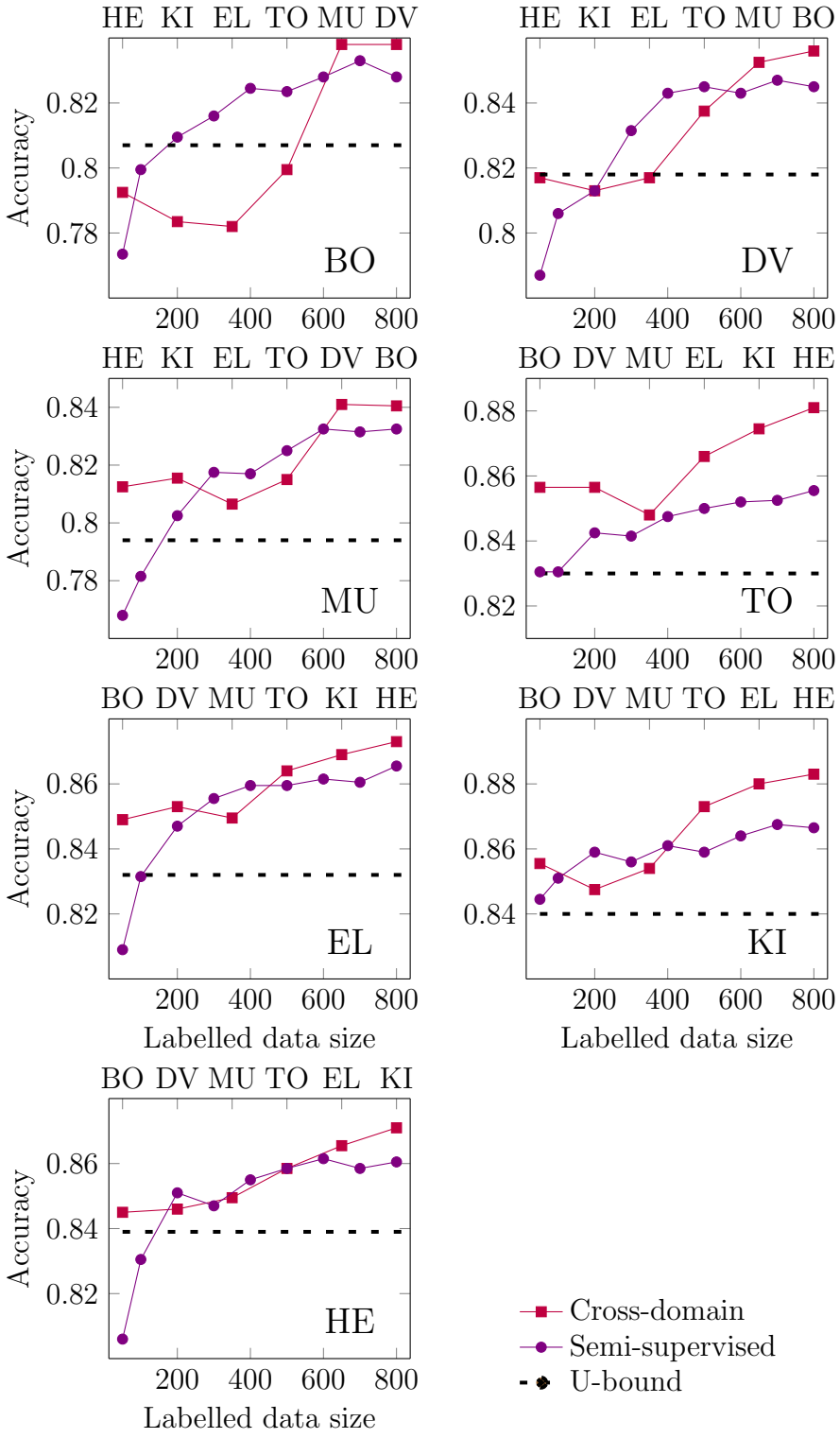

Figure 7.12: Comparison of the semi-supervised and cross-domain approaches (binary case).

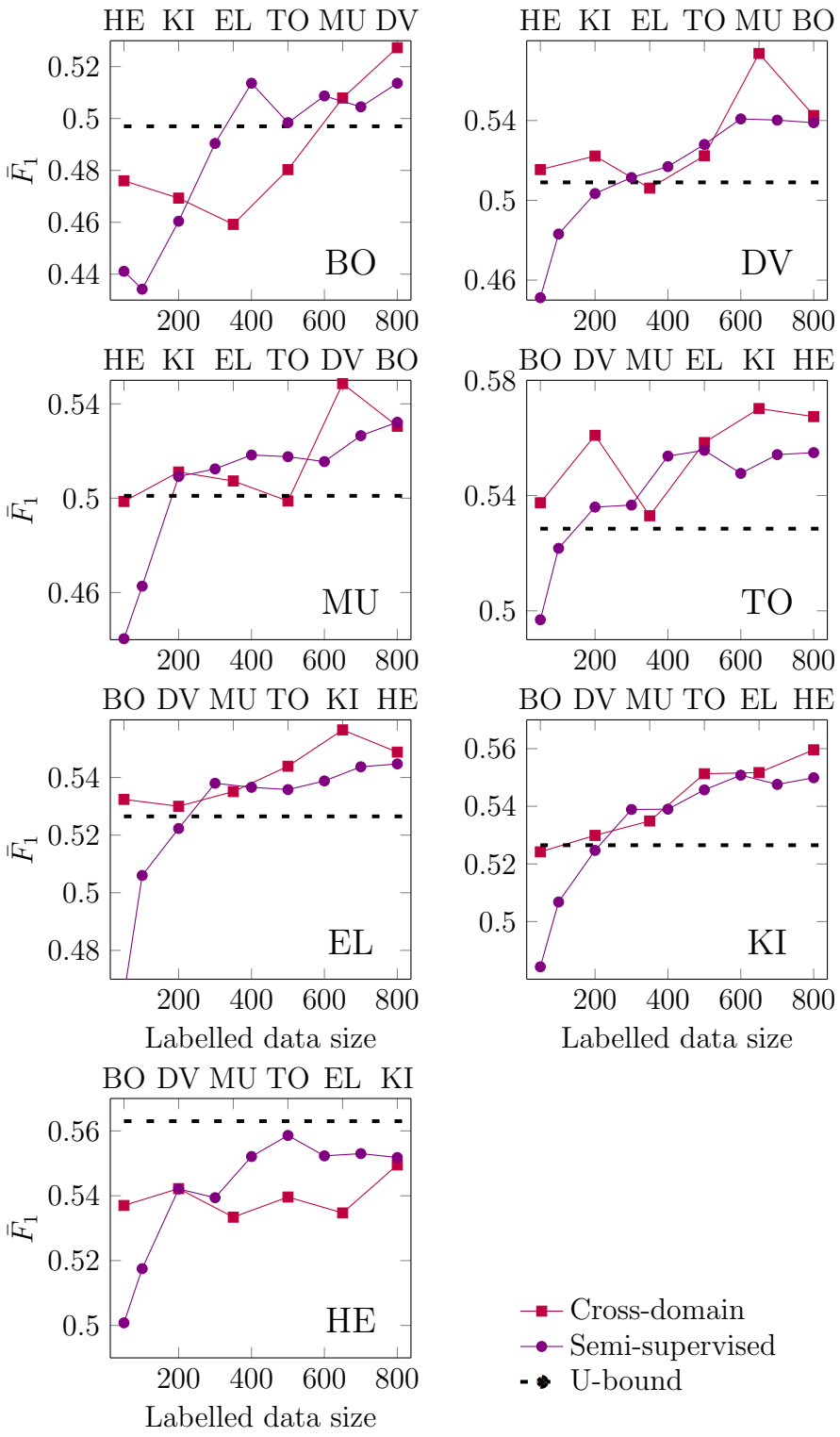

Figure 7.13: Comparison of the semi-supervised and cross-domain approaches (multiclass case).

Cross-domain learning for simple domains reaches the upper bound independently of the choice of source domain, which makes it preferable to semi-supervised learning. Interestingly, the cross-domain results for MU are similar to those for simple domains, which we think is due to the low upper bound for the MU domain. Moving the MU upper bound to the level of 0.81 or 0.82 achieved for BO and DV, would make the MU performance close to that of the other complex target domains. Cross-domain learning is also more beneficial for complex target domains when source-target domain pairs are similar. However, performance degrades for complex target domains when source and target data are different. For these data characteristics, semi-supervised learning with approximately 200 labelled documents should be used.

The multiclass results mostly repeat the binary results, although there are some differences that are worth noting (Figure 7.13). First, the cross-domain $\bar{F}_1$ values are quite high even for some complex domains. However, an additional study with more domains is required to prove the significance of these values. Second, the results for the HE domain are substantially lower than the upper bound. This is due to a very high upper bound for HE[5] as opposed to the poor performance of the graph-based algorithms for this domain because its $\bar{F}_1$ levels are comparable to those of TO, EL and KI. Moving the upper bound to the level of ≈ 0.53 would give us the same result as for the other simple domains. Finally, the number of labelled

[5] As mentioned in Chapter 3, we do not currently have an explanation for this.

examples needed for semi-supervised learning to reach the upper bound levels is approximately 300-400 documents and is higher than that for the binary case. This is an expected result due to the increased number of sentiment classes.

In light of these findings, we formulate the following recommendations for choosing the most beneficial approach in the framework of graph-based learning:

1. If the target domain is simple then cross-domain learning trained with any source data can be used.

2. If the target domain is complex and there is source data that is similar to the target data, then cross-domain learning trained on these similar data should be used.

3. If the target domain is complex and all source datasets are different, then semi-supervised learning trained with at least 300 labelled examples should be used.

It is worth emphasising that these recommendations are valid under certain conditions as determined by the scope of this thesis and our experimental results. For the graph-based learning algorithm, the best LP modification according to the learning setup and the number of sentiment classes should be used. For sentiment similarity, the hybrid similarity measure should be applied.

Chapter 8

Conclusions and future work

This chapter revisits the goals stated at the beginning of the thesis, summarises the findings established in this study and suggests directions for further research.

8.1 Goals revisited

The main aims of the thesis were to address the problem of limited availability of data in sentiment analysis and to advance research in semi-supervised and cross-domain approaches for sentiment classification, considering binary as well multiclass sentiment scales. These aims were met by achieving the three goals stated in Chapter 1.

The **first goal** was to explore ways of constructing sentiment graphs that allow the accurate estimation of sentiment similarity and, in addition, are easy to build, do not require deep linguistic analysis and do not involve manual annotation effort. This goal was achieved in Sections 4.2, 6.5 and 7.4.

In Section 4.2, two document representations to capture the sentiment strength of documents were suggested. The first, feature-based

representation, can be seen as domain-specific, while the second, unit-based representation, can be referred to as domain-independent. Both representations exploit sentiment lexicons, for which the SO-CAL dictionaries were used. To obtain document representations adjusted to the product review genre, the SO-CAL dictionaries were enhanced with new sentiment markers mined from a large corpus of Amazon reviews. Evaluation proved the SO-CAL adaptation to be effective for all domains under consideration. Although the SO-CAL dictionaries were updated by consulting the data, the repetition of this process is not required when working with other review data, as we expect the dictionaries to work well within the genre of product reviews.

The feature-based and unit-based document representations constructed were used to compute sentiment similarity. The intrinsic evaluation of different document representation components yielded two important results (Table 4.2). First, the hybrid similarity measure that incorporates both feature-based and unit-based representations performed best, which indicates that domain-specific and domain-independent information are both valuable for measuring sentiment. Second, the document representation components which have the highest impact on the similarity measure were identified. They include document features, the percentage of positive/negative words (PWP), the percentage of positive/negative sentences (PSP) and the percentage of positive/negative words in titles (TitlePWP).

The document representation components were also evaluated

extrinsically in semi-supervised and cross-domain settings (Sections 6.5 and 7.4). The semi-supervised evaluation revealed that domain-independent components make the results almost insensitive to the amount of labelled data used (Figures 6.11 and 6.12). In addition, for small labelled data sizes the similarity measure based solely on the unit-based representation performs either comparably to or significantly better than the hybrid similarity measure. According to the cross-domain evaluation results, the unit-based similarity measure appears to be independent of source domains as well. This quality could be crucial if the unit-based representation achieved the best results. However, due to its limitations, it is able to capture only simple sentiment phenomena. In future, we intend to enhance the unit-based representation by incorporating several important valence shifters (at the moment only negation is treated) similarly to Taboada et al. (2011).

Document features proved to capture sentiment better than document units, especially for the multiclass case (Figures 6.11 and 6.12). But due to the sparseness of the document-based representation, it provides a good estimation for sentiment similarity only when the amount of labelled data is relatively large. Document features are also more sensitive to source domains, showing better results when source and target domains are similar (Figure 7.10). According to both extrinsic evaluations, the hybrid similarity measure was the most successful for sentiment classification. Moreover, it benefits from the positive traits of both representations, performing well for small

and large labelled data sizes as well as for similar and dissimilar domain pairs.

The **second goal** of the thesis was to develop and evaluate a graph-based sentiment analysis system which can be used in semi-supervised and cross-domain settings and can handle multiclass classification. This goal was achieved in Section 3.2 and Chapters 4, 6 and 7.

The graph-based sentiment analysis system developed in this thesis comprises two modules: the preprocessing module, presented in Section 3.2, and the core of the system - the sentiment classification module - described in Chapter 4. The sentiment classification module (Figure 4.3) is designed in the framework of graph-based learning and implements the most well-known graph-based algorithm - LP. As input, it requires the similarity measure to be specified, which is further used for graph construction (see above). The graph-based inference stage of the module allows two modifications to the graph structure (prioritising either labelled or unlabelled neighbours and/or incorporating scores given by external classifiers) and the CMN normalisation of the output after each iteration. These modifications were implemented in three LP variants. During the preprocessing stage, the output probabilities can be modified using two normalisation techniques and/or the hierarchical probability combination rule.

The system was thoroughly evaluated in semi-supervised and cross-domain settings as described in Chapters 6 and 7. Our evaluation pursued the following five objectives.

The *first objective of the system evaluation* was to compare all LP modifications in different configurations to establish the most successful graph-based algorithm and its configuration. The semi-supervised and cross-domain experiments showed a marginal effect of modifications to the graph structure offered by the LP variants. In contrast, normalisation proved to be critical, leading to a significant improvement in the results. In particular, the LB normalisation is effective at the post-processing step, while CMN is more beneficial when used after each iteration (as in $RANK$). The $HIER$ probability combination rule yields a substantial gain in performance if the labelled data is sufficient.

To determine which algorithm performs equally well in semi-supervised and cross-domain settings, a comparative analysis of the results in both evaluation setups was conducted. For the binary case, LP_γ+LB gave the highest semi-supervised accuracy, while $RANK$ performed best in the cross-domain experiments. However, in semi-supervised settings LP_γ+LB considerably outperformed $RANK$, especially for small amounts of labelled data, whereas in cross-domain settings these methods gave comparable results. This makes LP_γ+LB the most beneficial overall.

In multiclass classification, semi-supervised and cross-domain approaches agreed to some extent on the best LP variant and its configuration. LP_γ+LB and $RANK$+$HIER$ performed best in semi-supervised settings, whereas LP_γ+$HIER$+LB and $RANK$+$HIER$ were better than other methods in cross-domain settings. The $HIER$ probability combination rule is beneficial

for the cross-domain performance of LP_γ (Figure 7.5). However, it is harmful for LP_γ in semi-supervised settings when less than 200 labelled examples are available (Figure 6.6). Since we assume that labelled data is very expensive to acquire, we consider LP_γ+LB to be the best in semi-supervised settings.

It is worth pointing out that there is no preference for either LP_γ+LB (or LP_γ+$HIER$+LB in cross-domain settings) or $RANK$+$HIER$, as they serve different purposes. While they both have comparable $\bar{F}_1$ values, the former yields very high $macroF_1$ values but quite moderate accuracy, whereas the latter shows the opposite results. Therefore, the choice of either LP_γ+LB or $RANK$+$HIER$ should be governed by the task requirements, as explained in Section 7.2.

The *second objective of the system evaluation* was to examine the sensitivity of the most successful LP modifications to variations of their parameters. In most cases, the algorithms were either insensitive to variations of some parameters (for example, all algorithms to variations of k_u, or LP_γ+LB to variations of β) or the dependency of their performance on parameter values was systematic (Figures 6.9, 6.10 and 7.9). Thus, the choice of a parameter value close to the optimal value of that parameter would guarantee accurate results.

The *third objective of the system evaluation* was to analyse the impact of different document representation components on the semi-supervised and cross-domain results. This was met by achieving the first goal of the thesis and was addressed above.

The *fourth objective of the system evaluation* was to study the dependency of graph-based algorithms on domain complexity and domain similarity. This was carried out whilst fulfilling the third goal of the thesis and will be discussed below.

Finally, the *fifth objective of the system evaluation* was to conduct a comparison of the best LP modifications with prominent existing semi-supervised and cross-domain approaches. The comparative analysis indicated the superiority of the graph-based algorithms over other methods in both semi-supervised and cross-domain settings. Besides their excellent performance, other advantages of the graph-based algorithms were identified:

1. The graph-based algorithms are much less sensitive to data characteristics than other methods. For example, semi-supervised reference methods, especially CO-TRAIN (Li et al., 2010a), appeared to be quite sensitive to data, showing a significant difference in accuracy between results on simple and complex domains (Table 6.4). In contrast, semi-supervised graph-based performance does not differ much from simple to complex domains. Moreover, the difference in performance is much smaller when the unit-based document representation is used to compute sentiment similarity. Concerning the cross-domain results, the performance of all the reference methods degraded drastically for dissimilar source and target domains (Figure 7.11). In contrast, the best LP variants showed a much smaller loss of accuracy, especially for simple target domains. Figure 7.10 implies that this low dependency of the graph-based results on source domains when

the target domain is simple is due to the positive impact of the unit-based components.

2. The graph-based algorithms are able to perform well even when limited labelled data is available. For semi-supervised settings, this was proved by our comparisons with the reference methods. However, this quality can also be important in cross-domain settings if source data is scarce. All the cross-domain reference methods rely on a large amount of labelled data from a source domain as they exploit co-occurrences between domain-specific and domain-independent sentiment markers. In contrast, the graph-based algorithms do not require much labelled data. As shown by the semi-supervised experiments, the benefit from additional labelled data decreases substantially when the labelled data size exceeds 500-600 examples (Figure 6.5). Although cross-domain experiments with a variable amount of labelled data were not carried out, we would expect a similar result in cross-domain settings.

3. The graph-based algorithms can easily handle both binary and multiclass classification. In contrast, most of reference methods were developed to tackle the binary classification problem and their extension to multiclass cases is either questionable or difficult to achieve.

We believe that these advantages provide solid support for the choice of graph-based learning as our main approach.

The **third goal** of the thesis was to undertake a comparison between semi-supervised and cross-domain approaches to develop recommendations

to help select the most pertinent approach given the data available. This goal was achieved in Chapter 5 and 7.

In Chapter 5, two data characteristics, domain similarity and domain complexity, were introduced and different functions for their estimation were suggested and examined. For domain complexity, two vocabulary richness measures, TTR and the percentage of rare words, were tested. The Pearson correlation between these measures and in-domain accuracies showed the advantage of TTR for estimating domain complexity (Table 5.1). Similar experiments were conducted to establish a function for estimating domain similarity, where seven similarity functions were assessed. As a result, two most accurate functions were found: χ^2 and Jensen-Shannon divergence (Table 5.2), which is in accordance with previous work on domain similarity for other NLP tasks. We also determined the boundaries (in terms of values of the suggested functions) between simple and complex domains, and between similar and different domains (Figures 5.1 and 5.2). However, an additional study is required to validate these boundaries for other datasets.

In Chapter 7, the cross-domain and semi-supervised performances were compared in terms of their best-performing algorithms, optimal parameter values and most accurate similarity measures. The main results of this comparison were summarised in the earlier part of this chapter. We also found that semi-supervised and cross-domain graph-based performances are partly determined by domain complexity and domain similarity. In particular, the semi-supervised experiments indicate that complex domains

need more labelled data to achieve a performance similar to that on simple domains (Figures 6.5 and 6.8). The cross-domain experiments demonstrate considerably better results when source and target data are similar (Figures 7.3 and 7.7). However, if target data is simple, the cross-domain performance remains higher than the upper bounds for most domain pairs. On the basis of these results, we developed a set of simple recommendations which rely on source and target domain characteristics to help choose between semi-supervised and cross-domain approaches (Section 7.7). These recommendations can be used when applying the best *LP* modification together with the hybrid similarity measure.

8.2 Original contributions

From the goals that were achieved in this thesis, we can summarise our findings into three main original contributions.

The **first original contribution** is the design of the sentiment similarity measure, which is unsupervised, easy to compute, does not require deep linguistic analysis and, most importantly, helps to achieve accurate classification results. Although, for the purposes of this thesis, the similarity measure was tailored to product review data, it can be easily adapted to other genres by excluding the TitlePWP component if titles are not available, and, if necessary, switching to different sentiment lexicons which are more relevant for the corresponding genre.

The **second original contribution** is the development, implementation

and evaluation of the graph-based sentiment analysis system that a) can cope with the challenges of limited data availability by using semi-supervised and cross-domain approaches, b) is able to perform multiclass classification, and c) achieves highly accurate results which are superior to those of most state-of-the-art semi-supervised and cross-domain systems.

The **third original contribution** is the joint and systematic analysis of the semi-supervised and cross-domain graph-based results and the development of recommendations for selecting the most pertinent learning approach given the data available in the framework of graph-based learning. The recommendations are based on two domain characteristics, domain similarity and domain complexity, which were shown to have a significant impact on the semi-supervised and cross-domain graph-based results.

8.3 Directions for future research

The scope of the research undertaken in this thesis is limited to specific graph-based algorithms and the product review domain. Therefore, one of the principal directions of our future work will be an extension of the scope to new methods and domains. In particular, we intend to implement and test two of the most recently proposed graph-based methods, modified adsorption (Talukdar and Crammer, 2009) and measure propagation (Subramanya and Bilmes, 2011), which present some advantages over LP. In addition, as mentioned in Chapter 4, various ways of transforming transductive learning

settings into inductive ones will be assessed (Zhu et al., 2003b; Chapelle et al., 2002; Delalleau et al., 2005; Sindhwani et al., 2005).

The hybrid similarity measure demonstrated a good estimate for sentiment similarity and largely determined the excellent performance of graph-based algorithms. However, its unit-based component can still be improved by a more accurate scoring function which assigns the final sentiment to a unit. As mentioned earlier in this chapter, we plan to incorporate important polarity shifters, such as intensifiers and irrealis grammatical moods (Taboada et al., 2011). The feature-based component can also be enhanced by the use of topic models in addition to document features. As shown by He et al. (2011), joint sentiment-topic (JST) models are beneficial for cross-domain sentiment classification. The analysis of polarity-bearing topics extracted by JST revealed the ability of JST to group words from different domains but bearing similar sentiment. This quality of JST may help to overcome the difference between source and target domain distributions.

The product review domain is a good starting point when researching a new approach for sentiment analysis as it is a well established field which has plenty of labelled data available. This eases comparisons between different methods. For example, in this thesis, the review domain helped us to conduct a fair comparison of the LP modifications with state-of-the-art approaches in semi-supervised and cross-domain settings. At the same time, automatic identification of review ratings may be considered not really

necessary, as in many environments, reviews are manually rated by users. The real world applications require ratings for other types of online data. The increasing popularity of social networks and microblogs make these social media streams a very important and valuable source of information as they contain immediate user responses to events and situations, as well as user predictions and speculation about future events. However, these social media streams lack labelled data, making it impossible to run supervised learning approaches. Mejova and Srinivasan (2012) showed that reviews can be suitable source data for sentiment classification of tweets. We intend to explore this further by running the graph-based algorithms designed in this thesis on the cross-genre task. We expect better results than those demonstrated in Mejova and Srinivasan (2012) as their classifier was not adapted to target domains.

None of the algorithms explored in this thesis rely on deep linguistic information. For this reason, they can be easily adapted to other languages. Therefore, one of the future research directions could be to perform sentiment classification for languages other than English.

Bibliography

Abbasi, A., Chen, H., and Salem, A. (2008). Sentiment analysis in multiple languages: Feature selection for opinion classification in Web forums. *ACM Transactions on Information Systems (TOIS)*, 26(3):1–34.

Andreevskaia, A. and Bergler, S. (2006). Mining WordNet for a fuzzy sentiment: Sentiment tag extraction from WordNet glosses. In *Proceedings of EACL*, pages 209–216.

Andreevskaia, A. and Bergler, S. (2008). When specialists and generalists work together: Overcoming domain dependence in sentiment tagging. In *Proceedings of ACL*, pages 290–298.

Artstein, R. and Poesio, M. (2008). Inter-coder agreement for computational linguistics. *Computational Linguistics*, 34(4):555–596.

Asch, V. V. and Daelemans, W. (2010). Using domain similarity for performance estimation. In *Proceedings of ACL Workshop on Domain Adaptation for Natural Language Processing*, pages 31–36.

Asur, S. and Huberman, B. A. (2010). Predicting the future with social media. In *Proceedings of IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology*, volume 1, pages 492–499.

Aue, A. and Gamon, M. (2005). Customizing sentiment classifiers to new domains: A case study. In *Proceedings of RANLP*.

Baayen, H. (2001). *Word Frequency Distributions*. Kluwer Academic Publishers.

Bai, X. (2011). Predicting consumer sentiments from online text. *Decision Support Systems*, 50(4):732–742.

Bal, B. K. (2014). Analyzing opinions and argumentation in news editorials and op-eds. *International Journal of Advanced Computer Science and Applications*, Special Issue on Natural Language Processing:22–29.

Balahur, A., Steinberger, R., Kabadjov, M., Zavarella, V., van der Goot, E., Halkia, M., Pouliquen, B., and Belyaeva, J. (2010). Sentiment analysis in the news. In *Proceedings of LREC*, pages 2216–2220.

Barbosa, L. and Feng, J. (2010). Robust sentiment detection on Twitter from biased and noisy data. In *Proceedings of COLING: Posters*, pages 36–44.

Belkin, M., Niyogi, P., and Sindhwani, V. (2006). Manifold regularization: A geometric framework for learning from labeled and unlabeled examples. *Journal of Machine Learning Research*, 7:2399–2434.

Bengio, Y., Delalleau, O., and Roux, N. L. (2006). *Semi-Supervised Learning*, chapter 11. Label Propagation and Quadratic Criterion, pages 193–216. The MIT Press.

Biber, D., Conrad, S., and Leech, G. (2002). *The Longman Student Grammar of Spoken and Written English*. Harlow: Longman.

Bilmes, J. and Subramanya, A. (2011). *Scaling up Machine Learning: Parallel and Distributed Approaches*, chapter 15. Parallel Graph-Based Semi-Supervised Learning, pages 307–330. Cambridge University Press.

Blei, D. M., Ng, A. Y., and Jordan, M. I. (2003). Latent dirichlet allocation. *Journal of Machine Learning Research*, 3:993–1022.

Blitzer, J., Dredze, M., and Pereira, F. (2007). Biographies, bollywood, boom-boxes and blenders: Domain adaptation for sentiment classification. In *Proceedings of ACL*, pages 440–447.

Blitzer, J., McDonald, R., and Pereira, F. (2006). Domain adaptation with structural correspondence learning. In *Proceedings of EMNLP*, pages 120–128.

Blum, A. and Chawla, S. (2001). Learning from labeled and unlabeled data using graph mincuts. In *Proceedings of ICML*, pages 19–26.

Blum, A. and Mitchell, T. (1998). Combining labeled and unlabeled data with co-training. In *Proceedings of COLT*, volume 98, pages 92–100.

Boiy, E. and Moens, M.-F. (2009). A machine learning approach to sentiment analysis in multilingual Web texts. *Information Retrieval*, 12(5):526–558.

Bollegala, D., Weir, D., and Carroll, J. (2011). Using multiple sources to construct a sentiment sensitive thesaurus for cross-domain sentiment classification. In *Proceedings of ACL*, pages 132–141.

Brody, S. and Elhadad, N. (2010). An unsupervised aspect-sentiment model for online reviews. In *Proceedings of HLT NAACL*, pages 804–812.

Carletta, J. (1996). Assessing agreement on classification tasks: The kappa statistic. *Computational Linguistics*, 22(2):249–254.

Carrillo de Albornoz, J., Plaza, L., Gervas, P., and Diaz, A. (2011). A joint model of feature mining and sentiment analysis for product review rating. In *Proceedings of ECIR*, pages 1820–1825.

Chang, C.-C. and Lin, C.-J. (2011). LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 2:1–27. Software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm.

Chapelle, O., Weston, J., and Schölkopf, B. (2002). Cluster kernels for semi-supervised learning. In *Proceedings of NIPS*, pages 585–592.

Chesley, P., Vincent, B., Xu, L., and Srihari, R. (2006). Using verbs and adjectives to automatically classify blog sentiment. In *AAAI Symposium on Computational Approaches to Analysing Weblogs (AAAI-CAAW)*, pages 27–29.

Cohen, J. (1960). A coefficient of agreement for nominal scales. *Educational and Psychological Measurement*, 20(1):37–46.

Cohn, D., Atlas, L., and Ladner, R. (1994). Improving generalization with active learning. *Machine Learning*, 15(2):201–221.

Conrad, J. G. and Schilder, F. (2007). Opinion mining in legal blogs. In *Proceedings of the International Conference on Artificial Intelligence and Law (ICAIL)*, pages 231–236.

Cortes, C. and Vapnik, V. (1995). Support-vector networks. *Machine Learning*, 20:273–297.

Cozman, F., Cohen, I., and Cirelo, M. (2003). Semi-supervised learning of mixture models. In *Proceedings of ICML*, pages 99–106.

Crammer, K., Kulesza, A., and Dredze, M. (2009). Adaptive regularization of weight vectors. *Advances in Neural Information Processing Systems*, 22:414–422.

Dasgupta, S. and Ng, V. (2009). Mine the easy, classify the hard: A semi-supervised approach to automatic sentiment classification. In *Proceedings of ACL/AFNLP*, pages 701–709.

Dave, K., Lawrence, S., and Pennock, D. M. (2003). Mining the peanut gallery: opinion extraction and semantic classification of product reviews. In *Proceedings of WWW*, pages 519–528.

Davidov, D., Tsur, O., and Rappoport, A. (2010). Enhanced sentiment learning using Twitter hashtags and smileys. In *Proceedings of COLING: Posters*, pages 241–249.

Delalleau, O., Bengio, Y., and Le Roux, N. (2005). Efficient non-parametric function induction in semi-supervised learning. In *Proceedings of the International Workshop on Artificial Intelligence and Statistics*, pages 96–103.

Devitt, A. and Ahmad, K. (2007). Sentiment polarity identification in financial news: A cohesion-based approach. In *Proceedings of ACL*, pages 984–991.

du Plessis, M. C. and Sugiyama, M. (2012). Semi-supervised learning of class balance under class-prior change by distribution matching. In *Proceedings of ICML*, pages 823–830.

Dunning, T. (1993). Accurate methods for the statistics of surprise and coincidence. *Computational Linguistics*, 19(1):61–74.

Dzeroski, S. and Zenko, B. (2004). Is combining classifiers with stacking better than selecting the best one? *Machine Learning*, 54:255–273.

Esuli, A. and Sebastiani, F. (2006a). Determining term subjectivity and term orientation for opinion mining. In *Proceedings of EACL*, pages 193–200.

Esuli, A. and Sebastiani, F. (2006b). SentiWordNet: A publicly available

lexical resource for opinion mining. In *Proceedings of LREC*, pages 417–422.

Forman, G. (2003). An extensive empirical study of feature selection metrics for text classification. *Journal of Machine Learning Research*, 3:1289–1305.

Gamon, M. (2004). Sentiment classification on customer feedback data: noisy data, large feature vectors, and the role of linguistic analysis. In *Proceedings of COLING*, pages 841–847.

Gamon, M. and Aue, A. (2005). Automatic identification of sentiment vocabulary: exploiting low association with known sentiment terms. In *Proceedings of ACL Workshop on Feature Engineering for Machine Learning in Natural Language Processing (FeatureEng)*, pages 57–64.

Glorot, X., Bordes, A., and Bengio, Y. (2011). Domain adaptation for large-scale sentiment classification: A deep learning approach. In *Proceedings of ICML*, pages 97–110.

Go, A., Bhayani, R., and Huang, L. (2009). Twitter sentiment classification using distant supervision. Technical report, Stanford University.

Goldberg, A. B. and Zhu, X. (2006). Seeing stars when there aren't many stars: graph-based semi-supervised learning for sentiment categorization. In *Proceedings of TextGraphs Workshop*, pages 45–52.

Guyon, I. and Elisseeff, A. (2003). An introduction to variable and feature selection. *Journal of Machine Learning Research*, 3:1157–1182.

Haimovitch, Y., Crammer, K., and Mannor, S. (2012). More is better: Large scale partially-supervised sentiment classication. *Journal of Machine Learning Research - Proceedings Track 25*, pages 175–190.

Hassan, A. and Radev, D. R. (2010). Identifying text polarity using random walks. In *Proceedings of ACL*, pages 395–403.

Hatzivassiloglou, V. and McKeown, K. R. (1997). Predicting the semantic orientation of adjectives. In *Proceedings of EACL*, pages 174–181.

He, Y., Lin, C., and Alani, H. (2011). Automatically extracting polarity-bearing topics for cross-domain sentiment classification. In *Proceedings of ACL*, pages 123–131.

Hinton, G. E. and Salakhutdinov, R. R. (2006). Reducing the dimensionality of data with neural networks. *Science*, 313(5786):504–507.

Jiang, J. (2008). A literature survey on domain adaptation of statistical classifiers.

Joachims, T. (2003). Transductive learning via spectral graph partitioning. In *Proceedings of ICML*, pages 290–297.

John, G. H., Kohavi, R., and Pfleger, K. (1994). Irrelevant features and the subset selection problem. In *Proceedings of ICML*, pages 121–129.

Kamps, J. and Marx, M. (2002). Words with attitude. In *Proceedings of the International Conference on Global WordNet*, pages 332–341.

Kamps, J., Marx, M., Mokken, R. J., and de Rijke, M. (2004). Using WordNet to measure semantic orientation of adjectives. In *Proceedings of LREC*, volume IV, pages 1115–1118.

Kennedy, A. and Inkpen, D. (2006). Sentiment classification of movie reviews using contextual valence shifters. *Computational Intelligence: Special Issue on Sentiment Analysis*, 22(2):110–125.

Kilgarriff, A. (2001). Comparing corpora. *International Journal of Corpus Linguistics*, 6(1):97–133.

Kim, J., Li, J.-J., and Lee, J.-H. (2009). Discovering the discriminative views: Measuring term weights for sentiment analysis. In *Proceedings of ACL/AFNLP*, pages 253–261.

Kim, S.-M. and Hovy, E. (2004). Determining the sentiment of opinions. In *Proceedings of COLING*, pages 1367–1373.

Kim, S.-M. and Hovy, E. (2006). Identifying and analyzing judgment opinions. In *Proceedings of HLT-NAACL*, pages 200–207.

Kouloumpis, E., Wilson, T. A., and Moore, J. (2011). The good the bad and the OMG! In *Proceedings of ICWSM*, pages 538–541.

Kullback, S. and Leibler, R. A. (1951). On information and sufficiency. *The Annals of Mathematical Statistics*, 22(1):79–86.

Li, F., , Pan, S. J., Jin, O., Yang, Q., and Zhu, X. (2012). Cross-domain co-extraction of sentiment and topic lexicons. In *Proceedings of ACL*, pages 410–419.

Li, F., Liu, N., Jin, H., Zhao, K., Yang, Q., and Zhu, X. (2011). Incorporating reviewer and product information for review rating prediction. In *Proceedings of IJCAI*, pages 1820–1825.

Li, S., Huang, C.-R., Zhou, G., and Lee, S. Y. M. (2010a). Employing personal/impersonal views in supervised and semi-supervised sentiment classification. In *Proceedings of ACL*, pages 414–423.

Li, S., Lee, S. Y. M., Chen, Y., Huang, C.-R., and Zhou, G. (2010b). Sentiment classification and polarity shifting. In *Proceedings of COLING*, pages 635–643.

Li, S. and Zong, C. (2008). Multi-domain sentiment classification. In *Proceedings of ACL:Short Papers*, pages 257–260.

Lin, C. and He, Y. (2009). Joint sentiment/topic model for sentiment analysis. In *Proceedings of CIKM*, pages 375–384.

Lin, J. (1991). Divergence measures based on the Shannon entropy. *IEEE Transactions on Information Theory*, 37(1):145–151.

Liu, B. (2012). *Sentiment Analysis and Opinion Mining*. Morgan&Claypool Publishers.

Liu, H., Lieberman, H., and Selker, T. (2003). A model of textual affect sensing using real-world knowledge. In *Proceedings of IUI*, pages 125–132.

Maks, I. and Vossen, P. (2013). Sentiment analysis of reviews: Should we analyze writer intentions or reader perceptions? In *Proceedings of RANLP*, pages 415–419.

Manning, C. D., Raghavan, P., and Schutze, H. (2008). *Introduction to Information Retrieval*. Cambridge University Press.

Martin, J. and White, P. R. R. (2005). *The Language of Evaluation: Appraisal in English*. Palgrave, London.

Martineau, J. and Finin, T. (2009). Delta TFIDF: An improved feature space for sentiment analysis. In *Proceedings of ICWSM*, pages 258–261.

McDonald, R., Hannan, K., Neylon, T., Wells, M., and Reynar, J. (2007). Structured models for fine-to-coarse sentiment analysis. In *Proceedings of ACL*, pages 432–439.

Mei, Q., Ling, X., Wondra, M., Su, H., and Zhai, C. (2007). Topic sentiment mixture: Modeling facets and opinions in weblogs. In *Proceedings of WWW*, pages 171–180.

Mejova, Y. and Srinivasan, P. (2012). Crossing media streams with sentiment: Domain adaptation in blogs, reviews and Twitter. In *Proceedings of ICWSM*, pages 234–241.

Miller, G. A. (1995). A lexical database for English. *Communications of the ACM*, 38(11):39–41.

Mishne, G. and Glance, N. (2006). Predicting movie sales from blogger sentiment. In *AAAI Symposium on Computational Approaches to Analysing Weblogs (AAAI-CAAW)*, pages 155–158.

Mladenic, D. and Grobelnik, M. (1999). Feature selection for unbalanced class distribution and Naive Bayes. In *Proceedings of ICML*, pages 258–267.

Mullen, T. and Collier, N. (2004). Sentiment analysis using support vector machines with diverse information sources. In *Proceedings of EMNLP*, pages 412–418.

Ng, A., Jordan, M., and Weiss, Y. (2001). On spectral clustering: Analysis and an algorithm. In *Advances of NIPS*, pages 849–856.

Ng, V., Dasgupta, S., and Arifin, S. M. N. (2006). Examining the role of linguistic knowledge sources in the automatic identification and classification of reviews. In *Proceedings of COLING/ACL: Poster*, pages 611–618.

Nigam, K., Mccallum, A. K., Thrun, S., and Mitchell, T. (2000). Text classification from labeled and unlabeled documents using EM. *Machine Learning - Special issue on information retrieval*, 39(2-3):103–134.

Osgood, C. E., Suci, G., and Tannenbaum, P. (1957). *The Measurement of Meaning*. University of Illinois Press.

Pak, A. and Paroubek, P. (2010). Twitter as a corpus for sentiment analysis and opinion mining. In *Proceedings of LREC*.

Paltoglou, G. and Thelwall, M. (2010). A study of information retrieval weighting schemes for sentiment analysis. In *Proceedings of ACL*, pages 1386–1395.

Paltoglou, G. and Thelwall, M. (2012). Seeing stars of valence and arousal in blog posts. *Journal of IEEE Transactions of Affective Computing*, 4(1):116–123.

Pan, S. J., Ni, X., Sun, J.-T., Yang, Q., and Chen, Z. (2010). Cross-domain sentiment classification via spectral feature alignment. In *Proceedings of WWW*, pages 751–760.

Pan, S. J., Tsang, I. W., Kwok, J. T., and Yang, Q. (2011). Domain adaptation via transfer component analysis. *IEEE Transactions on Neural Networks*, 22(2):199–210.

Pang, B. and Lee, L. (2004). A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts. In *Proceedings of ACL*, pages 271–278.

Pang, B. and Lee, L. (2005). Seeing stars: Exploiting class relationships for

sentiment categorization with respect to rating scales. In *Proceedings of ACL*, pages 115–124.

Pang, B. and Lee, L. (2008). Opinion mining and sentiment analysis. *Foundations and Trends in Information Retrieval*, 2(1-2):1–135.

Pang, B., Lee, L., and Vaithyanathan, S. (2002). Thumbs up? Sentiment classification using machine learning techniques. In *Proceedings of EMNLP*, pages 79–86.

Park, S., Lee, K., and Song, J. (2011). Contrasting opposing views of news articles on contentious issues. In *Proceedings of ACL*, pages 340–349.

Plank, B. and van Noord, G. (2011). Effective measures of domain similarity for parsing. In *Proceedings of ACL*, pages 1566–1576.

Poesio, M. and Artstein, R. (2005). The reliability of anaphoric annotation, reconsidered: Taking ambiguity into account. In *Proceedings of the Workshop on Frontiers in Corpus Annotation II: Pie in the Sky*, pages 76–83.

Polanyi, L. and Zaenen, A. (2006). *Contextual Valence Shifters*, volume 20 of *The Information Retrieval Series*, chapter 1, pages 1–10. Springer Netherlands.

Prabowo, R. and Thelwall, M. (2009). Sentiment analysis: A combined approach. *Journal of Informetrics*, 3(1):143 – 157.

Read, J. (2005). Using emoticons to reduce dependency in machine learning techniques for sentiment classification. In *Proceedings of the ACL Student Research Workshop*, pages 43–48.

Read, J. and Carroll, J. (2009). Weakly supervised techniques for domain-independent sentiment classification. In *Proceedings of CIKM workshop on Topic-sentiment analysis for mass opinion (TSA)*, pages 45–52.

Renyi, A. (1961). On measures of information and entropy. In *Proceedings of the Berkeley Symposium on Mathematics, Statistics and Probability*, volume 1, pages 547–561.

Riloff, E., Patwardhan, S., and Wiebe, J. (2006). Feature subsumption for opinion analysis. In *Proceedings of EMNLP*, pages 440–448.

Sadikov, E., Parameswaran, A. G., and Venetis, P. (2009). Blogs as predictors of movie success. In *Proceedings of ICWSM*, pages 304–307.

Saerens, M., Patrice, M., and Decaestecker, C. (2001). Adjusting the outputs of a classifier to new a priori probabilities: A simple procedure. *Neural Computation*, 14:21–41.

Scheible, C., Laws, F., Michelbacher, L., and Schütze, H. (2010). Sentiment translation through multi-edge graphs. In *Proceedings of COLING: Posters*, pages 1104–1112.

Scott, W. (1955). Reliability of content analysis: The case of nominal scale coding. *Public Opinion Quarterly*, 19(3):321–325.

Seki, Y., Evans, D. K., Ku, L.-W., Sun, L., Chen, H.-H., and Kando, N. (2008). Overview of multilingual opinion analysis task at NTCIR-7. In *Proceedings of the NTCIR-7 Workshop*, pages 185–203.

Sindhwani, V., Niyogi, P., and Belkin, M. (2005). Beyond the point cloud: from transductive to semi-supervised learning. In *Proceedings of ICML*, pages 824–831.

Speriosu, M., Sudan, N., Upadhyay, S., and Baldridge, J. (2011). Twitter polarity classification with label propagation over lexical links and the follower graph. In *Proceedings of EMNLP*, pages 53–63.

Srihari, R. K., Inc, C., Li, W., Niu, C., and Cornell, T. (2003). Infoxtract: A customizable intermediate level information extraction engine. *SEALTS '03 Proceedings of the HLT-NAACL 2003 workshop on Software engineering and architecture of language technology systems*, 8:51–58.

Stone, P. J., Dunphy, D. C., Smith, M. S., Ogilvie, D. M., and associates (1966). *The General Inquirer: A Computer Approach to Content Analysis*. MIT.

Strapparava, C. and Valitutti, A. (2004). WordNet-Affect: An affective extension of WordNet. In *Proceedings of LREC*, pages 1083–1086.

Subramanya, A. and Bilmes, J. (2011). Semi-supervised learning with

measure propagation. *Journal of Machine Learning Research*, 12:3311–3370.

Taboada, M., Brooke, J., Tofiloski, M., Voll, K., and Stede, M. (2011). Lexicon-based methods for sentiment analysis. *Computational Linguistics*, 37(2):267–307.

Taboada, M. and Grienve, J. (2004). Analyzing appraisal automatically. In *Proceedings of the AAAI Symposium on Exploring Attitude and Affect in Text: Theories and Applications*, pages 158–161.

Talukdar, P. P. and Crammer, K. (2009). New regularized algorithms for transductive learning. In *Proceedings of ECML PKDD*, pages 442–457.

Tan, S., Wu, G., Tang, H., and Cheng, X. (2007). A novel scheme for domain-transfer problem in the context of sentiment analysis. In *Proceedings of CIKM*, pages 979–982.

Tesitelova, M. (1992). *Quantitative Linguistics*. Academia.

Thelwall, M., Buckley, K., and Paltoglou, G. (2012). Sentiment strength detection for the social web. *Journal of the American Society for Information Science and Technology*, 63(1):163–173.

Thelwall, M., Buckley, K., Paltoglou, G., Cai, D., and Kappas, A. (2010). Sentiment strength detection in short informal text. *Journal of the American Society for Information Science and Technology*, 61(12):2544–2558.

Titov, I. and McDonald, R. (2008). Modeling online reviews with multi-grain topic models. In *Proceeding of WWW*, pages 111–120.

Turney, P. (2002). Thumbs up or thumbs down? Semantic orientation applied to unsupervised classification of reviews. In *Proceedings of ACL*, pages 417–424.

Turney, P. D. and Littman, M. L. (2003). Measuring praise and criticism: Inference of semantic orientation from association. *ACM Transactions on Information Systems (TOIS)*, 21(4):315–346.

van Rijsbergen (1975). *Information Retrieval*. London, UK: Butterworths.

Vapnik, V. (1998). *Statistical Learning Theory*. Wiley-Interscience.

Vincent, P., Larochelle, H., Bengio, Y., , and Manzagol, P.-A. (2008). Extracting and composing robust features with denoising autoencoders. In *Proceedings of ICML*, pages 1096–1103.

Whitelaw, C., Garg, N., and Argamon, S. (2005). Using appraisal groups for sentiment analysis. In *Proceedings of CIKM*, pages 625–631.

Wiebe, J. M. and Riloff, E. (2005). Creating subjective and objective sentence classifiers from unannotated texts. In *Proceedings of CICLing*, pages 486–497.

Wilson, T., Wiebe, J., and Hoffmann, P. (2005). Recognizing contextual

polarity in phrase-level sentiment analysis. In *Proceedings of HLT-EMNLP*, pages 347–354.

Wilson, T., Wiebe, J., and Hoffmann, P. (2009). Recognizing contextual polarity: An exploration of features for phrase-level sentiment analysis. *Computational Linguistics*, 35(3):399–433.

Wilson, T. A. (2008). *Fine-grained Subjectivity and Sentiment Analysis: Recognizing the intensity, polarity, and attitudes of private states*. PhD thesis, University of Pittsburgh.

Wu, Q., Tan, S., and Cheng, X. (2009). Graph ranking for sentiment transfer. In *Proceedings of ACL-IJCNLP: Short Papers*, pages 317–320.

Xu, G., Meng, X., and Wang, H. (2010). Build Chinese emotion lexicons using a graph-based algorithm and multiple resources. In *Proceedings of COLING*, pages 1209–1217.

Yang, P., Gao, W., Tan, Q., and Wong, K.-F. (2012). Information-theoretic multi-view domain adaptation. In *Proceedings of ACL*, pages 270–274.

Yang, Y. and Pedersen, J. O. (1997). A comparative study on feature selection in text categorization. In *Proceedings of ICML*, pages 412–420.

Yarowsky, D. (1995). Unsupervised word sense disambiguation rivaling supervised methods. In *Proceedings of ACL*, pages 189–196.

Yessenalina, A., Choi, Y., and Cardie., C. (2010). Automatically generating annotator rationales to improve sentiment classification. In *Proceedings of ACL*, pages 336–341.

Zagibalov, T. (2010). *Unsupervised and Knowledge-poor Approaches to Sentiment Analysis*. PhD thesis, University of Sussex.

Zagibalov, T. and Carroll, J. (2008). Automatic seed word selection for unsupervised sentiment classification of Chinese text. In *Proceedings of COLING*, pages 1073–1080.

Zaidan, O., Eisner, J., and Piatko, C. (2007). Using "Annotator Rationales" to improve machine learning for text categorization. In *Proceedings of NAACL HLT*, pages 260–267.

Zhao, X., Jiang, J., Yan, H., and Li, X. (2010). Jointly modeling aspects and opinions with a MaxEnt–LDA hybrid. In *Proceedings of EMNLP*, pages 56–65.

Zhou, D., Bousquet, O., Lal, T., Weston, J., and Scholkopf, B. (2004). Learning with local and global consistency. In *Proceedings of NIPS*, pages 321–328.

Zhou, S., Chen, Q., and Wang, X. (2013). Active deep learning method for semi-supervised sentiment classification. *Neurocomputing*, 120:536–546.

Zhu, X. (2005). *Semi-Supervised Learning with Graphs*. PhD thesis, Carnegie Mellon University.

Zhu, X. (2008). Semi-supervised learning literature survey. Technical report, University of Wisconsin Madison.

Zhu, X. and Ghahramani, Z. (2002). Learning from labeled and unlabeled data with label propagation. Technical report, Carnegie Mellon University.

Zhu, X., Ghahramani, Z., and Lafferty, J. (2003a). Semi-supervised learning using Gaussian fields and harmonic functions. In *Proceedings of ICML*, pages 912–919.

Zhu, X., Lafferty, J., and Ghahramani, Z. (2003b). Semi-supervised learning: From Gaussian fields to Gaussian processes. Technical report, Carnegie Mellon University.

Appendix A

Identified sentiment markers

Positive sentiment markers

No	word	LR	valence	domains
1	highly	4.3	3	BO, DV, MU, EL, KI, TO, HE
2	family	2.4	2	BO, DV, MU, KI, TO, HE
3	fast	2.3	2	BO, MU, EL, KI, HE
4	storage	2.2	1	EL, KI, TO
5	overall	2.2	2	MU, EL, KI, TO, HE
6	collection	2.2	1	DV, MU, TO
7	thanks	2.1	2	BO, DV, HE
8	combine	2.1	2	BO, DV, MU
9	include	2.1	2	DV, MU, KI
10	especially	2.1	2	MU, TO, HE
11	ride	2.1	1	DV, MU, TO
12	bring	2.0	1	BO, DV, MU, KI, TO
13	discover	2.0	2	BO, DV, MU
14	provide	2.0	2	BO, DV, MU
15	job	2.0	1	BO, DV, MU, EL, HE

16	addition	2.0	1	BO, DV, MU, KI, TO
17	definitely	1.9	2	BO, DV, MU, KI, TO, HE
18	simple	1.9	2	BO, DV, EL, KI, TO, HE
19	price	1.9	2	DV, MU, EL, KI, HE

Negative sentiment markers

No	word	LR	valence	domains
1	refund	-18.2	-4	EL, KI, TO, HE
2	return	-10.2	-4	EL, KI, TO, HE
3	buy_NOT	-6.2	-4	BO, DV, MU, EL, KI, TO, HE
4	work_NOT	-4.2	-4	DV, EL, KI, TO, HE
5	send	-3.6	-2	EL, KI, TO, HE
6	even_NOT	-3.5	-4	BO, DV, MU, EL, KI, TO, HE
7	contact	-3.3	-2	EL, KI, HE
8	make_NOT	-3.1	-3	BO, DV, TO
9	customer	-3.1	-3	EL, KI, HE
10	manufacturer	-3.1	-3	EL, KI, TO, HE
11	money	-2.9	-3	BO, DV, MU, EL, KI, TO, HE
12	stop	-2.8	-3	EL, KI, TO, HE
13	suppose	-2.7	-1	BO, DV, MU, KI, TO, HE
14	break	-2.7	-3	EL, KI, TO, HE
15	unless	-2.6	-1	BO, DV, MU, EL, KI, TO, HE

16	call	-2.5	-1	EL, KI, TO, HE
17	avoid	-2.5	-2	DV, MU, EL, KI
18	fix	-2.5	-2	EL, KI, TO
19	claim	-2.5	-2	BO, EL, HE
20	total	-2.5	-1	BO, DV, TO, HE
21	save	-2.4	-1	BO, DV, MU, KI, TO, HE
22	throw	-2.4	-2	BO, EL, KI, TO, HE
23	idea	-2.4	-1	EL, KI, TO, HE
24	service	-2.4	-1	EL, KI, HE
25	company	-2.3	-1	EL, KI, TO, HE
26	do_NOT	-2.2	-2	DV, MU, EL, TO, HE
27	none	-2.2	-2	BO, DV, TO
28	basically	-2.2	-1	BO, MU, EL, TO, HE
29	later	-2.2	-1	KI, TO, HE
30	stick	-2.1	-1	BO, MU, KI, TO, HE
31	cause	-2.1	-1	EL, KI, TO
32	replacement	-2.1	-3	EL, KI, TO, HE
33	again	-2.1	-2	EL, KI, TO
34	charge	-2.1	-2	EL, KI, TO
35	plastic	-2.1	-1	EL, KI, TO, HE
36	replace	-2.1	-3	EL, KI, TO
37	try	-2.0	-1	MU, EL, KI, TO

38	instead	-2.0	-2	BO, DV, MU, TO, HE
39	guess	-2.0	-1	BO, DV, EL, KI
40	dollar	-2.0	-3	EL, KI, TO
41	back	-2.0	-2	EL, KI, TO, HE
42	happen	-2.0	-1	EL, KI, TO, HE
43	sell	-1.9	-1	BO, MU, EL, KI, TO, HE
44	$	-1.9	-3	BO, DV, MU, TO, HE
45	either	-1.9	-1	BO, DV, EL, KI, TO, HE
46	maybe	-1.9	-1	BO, DV, MU, EL, KI, TO, HE

Appendix B

Amazon Review Coding Instructions

Please read and follow the instructions carefully

B.1 Introduction

We would like to investigate whether ratings to products given by Amazon customers in general are conformed to their actual reviews. Our interest is due to several reasons. First, we are interested to explore whether humans are able to differentiate between 1* and 2* as well as between 4* and 5* reviews. And, second, we want to estimate the maximum accuracy that automatic computer programs are able to achieve on these data.

The data is compound by 400 randomly sampled Amazon reviews on 4 topics: books, electronics, kitchen appliances and DVDs. Each topic contains 100 reviews. Only reviews rated with 1*, 2*, 4* and 5* have been selected. The number of reviews with the same number of stars can vary from topic to topic.

Having only the text of a review you will need to guess the number of stars its author gave to the described product.

B.2 Filling in the annotation form

You will be given the txt-file where all reviews are listed one by one. The reviews are grouped under the same topic. Each topic contains 100 reviews, where number of reviews rated as 1*, 2*, 4* and 5* can be different. Please record your judgement on product rating entering just a number in an allocated space under the review. Please do not leave this field empty, try to make a judgement even if the sentiment is unclear. Your concerns and doubts can be left in an allocated space under your judgement in the comment box. Although there are no 3* reviews in the data you can use 3* in a very rare case when, on your opinion, no other rating is appropriate. Please do not share your answers with anyone else or discuss your answers with anyone else.

B.3 Rating judgements

Below are descriptions of the judgements for you to make. There are no formal criteria for these judgements because we don't know any formal criteria for them! We want you to use your human knowledge and intuition to make your decisions.

- 1* - The product is described as very bad, very disappointing or very unsatisfactory.

- 2* - The user is not satisfied with the product, but it is not the worst

thing ever. There may be some overlap with the sentiments expressed in 1* reviews, but the sentiments tend to be less strong.

- 3* - This is used to indicate that you cannot decide on a judgement for a review (please note that there are no 3* reviews included in the file). Try not to use this if possible. Use sparingly, and ONLY in cases where:

    1. there is not enough information to distinguish the product under review (e.g. the user discusses more than 1 product and you cannot tell which is the one actually being sold);
    2. the review consists of equal amounts of positive and negative points and you cannot get a feel for which is more prominent;
    3. the review is a summary of the product with no user opinions and you cannot get a feel for an overall positive or negative sentiment.

- 4* - The product is described as good or very good; the user is generally satisfied with the product and likes it. There MAY be one or two negative points listed, but overall the user is happy. There may be some overlap with sentiments expressed in reviews judged as 5*, but the sentiments tend to be less strong.

- 5* - The product is described as very good or excellent; the user is very satisfied and is happy to recommend it to others.

When making judgements, please be as consistent with your previous decisions as possible.

Appendix C

Previously published work

1. **Natalia Ponomareva**, Mike Thelwall (2012) "Biographies or Blenders? Which Resource is Best for Cross-domain Sentiment Classification". *In Proceedings of the Conference on Intelligent Text Processing and Computational Linguistics (CICLing'12)*, Lecture Notes in Computer Science, Springer-Verlag, pp. 488-499.

2. **Natalia Ponomareva**, Mike Thelwall (2012) "Do neighbours help? An Exploration of Graph-based Algorithms for Cross-domain Sentiment Classification". *In Proceedings of the Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL'12)*, Association for Computational Linguistics, pp. 655-665.

3. **Natalia Ponomareva**, Mike Thelwall (2013) "Semi-supervised vs. Cross-domain Graphs for Sentiment Analysis". *In Proceedings of the Conference on Recent Advances in Natural Language Processing (RANLP'13)*, pp. 571-578.